

(12)

LEVEL 2

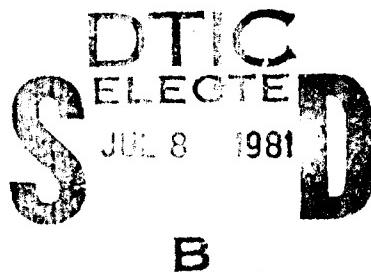
HDL-TM-80-15

December 1980

AD 101 134

A Detailed Review of the TACWAR Model

by John C. Ingram



U.S. Army Electronics Research
and Development Command
Harry Diamond Laboratories
Adelphi, MD 20783



Approved for public release. Distribution unlimited.

8 8 08 014

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturers' or trade names does not constitute an official indorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

A preliminary version of this report
was issued as HDL-PRL-80-10 in March 1980.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER HDL-TM-80-15	2. GOVT ACCESSION NO. AD-A101134	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and subtitle) A Detailed Review of the TACWAR Model	5. TYPE OF REPORT & PERIOD COVERED Technical Memorandum	
	6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s) John C. Ingram	8. CONTRACT OR GRANT NUMBER(s)	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Harry Diamond Laboratories 2800 Powder Mill Road Adelphi, MD 20783	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Ele: 65108A	
11. CONTROLLING OFFICE NAME AND ADDRESS Commander US Army Training and Doctrine Command Fort Monroe, VA 23561	12. REPORT DATE December 1980	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	13. NUMBER OF PAGES 77	
	15. SECURITY CLASS. (of this report) UNCLASSIFIED	
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES DRCMS Code: 66570808A9870 DA Project: 6A665708A987 HDL Project: E779E2		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) TACWAR Combat simulation Command, Control, and Communications		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A review of the TACWAR theater-level combat simulation model has been completed as an initial prerequisite to the primary objective of augmenting TACWAR to include Command, Control, and Communications Degradation (C ³ /D) considerations. This review has revealed a number of logic and programming "bugs" in the model that could also have remained undetected in other versions of the model throughout the TACWAR community of interest. It is the		

163050

J013

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. ABSTRACT (Cont'd)

- intent of the author to help the community eliminate these potential errors by cataloging them point by point and by recommending corrections.

UNCLASSIFIED

2 SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

CONTENTS

	<u>Page</u>
1. INTRODUCTION	5
2. SCOPE	5
3. SYSTEM COMPATIBILITY	6
4. DOCUMENTATION	6
5. ANALYSIS BY TACWAR PROGRAM SECTION	6
DISTRIBUTION	77

S DTIC
ELECTED
JUL 8 1981
B D

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/ _____	
Availability Codes	
Dist	Avail and/or
	Special
A	

1. INTRODUCTION

This report documents a number of errors (or potential errors) discovered in the TACWAR theater-level combat simulation model developed by the Institute for Defense Analysis (IDA) for the Department of Defense Studies Analysis and Gaming Agency (DoD-SAGA) and maintained by the Command and Control Technical Center (CCTC) of the Defense Communications Agency (DCA).

Since TACWAR is being considered a prime candidate to provide the net assessment results of the Army's Theater Nuclear Forces Survivability (TNF/S) program, the author, who is currently a participant in TNF/S, obtained a copy of version 2.1 TACWAR, including the baseline data base, from CCTC in June 1978. The principal area of interest to the author, as a member of the team responsible for including Command, Control, and Communications Degradation (C³/D) effects in TNF/S, has been to determine those portions of the TACWAR program that are amenable to C³/D and to augment the TACWAR program code, to reflect this inclusion of C³/D effects.* Because this effort requires an in-depth study and analysis of much of the TACWAR code, a simultaneous review of the existing code has been performed, revealing possible discrepancies, logic errors, and programming "bugs."

In documenting these errors it is not the intent of the author to try in any way to discredit or impugn the integrity or competence of the individuals who developed TACWAR. The author is well aware that any program the size of TACWAR is exceedingly difficult to debug, except through extensive use by a large community of interest, preferably working independently, but maintaining close communications on relevant matters of common interest. In fact, some of the "errors" discussed in this memorandum may simply reflect an incomplete understanding by the author of the intent of specific sections of the code. For such situations, the author would be grateful to be informed of correct interpretation, either formally or informally.

2. SCOPE

This document does not purport to include an exhaustive list of all errors or potential errors in TACWAR. Indeed, some sections of TACWAR and some subroutines have not yet been analyzed in detail by the author, either because they contain complex algorithms whose analysis requires documents not currently available to the author or because they have little or no C³/D aspect. Moreover, no attempt has been made to analyze the effect that the errors might have on the validity of the results obtained from TACWAR. It is quite possible that some errors produce almost trivial perturbations while others might be significant. In either case, however, it is recommended that, if an error exists, it be corrected regardless of the scope of its effect on the results.

*C³ modified TACWAR model, John C. Ingram, Harry Diamond Laboratories preliminary report (HDL-PRL-80-8, January 1980), to be published as HDL-TR-1918.

Specifically excluded from detailed analysis are the large subroutines DAMEVL and CEMDAM (formerly CHEMDAM), and their ancillary subroutines, used to evaluate the damage to targets by nuclear and chemical strikes. In addition, the entire "supp'y" model has not yet been analyzed in detail. Lastly, within many of the remaining subroutines the "output" statements (i.e., WRITE and FORMAT) have not yet been analyzed.

3. SYSTEM COMPATIBILITY

After receiving the original TACWAR program and data base from CCTC (Honeywell compatible version), the author converted the subroutines comprising the program to be compatible with the FORTRAN Extended compiler operating under OS-VS2 on the IBM/370-168 system available at the U.S. Army Harry Diamond Laboratories (HDL). This compatibility task necessitated the removal of the "PARAMETER" option available for the original Honeywell version. Other aspects of the conversion process, such as replacing the CHARACTER statements with techniques performing equivalent functions and recognizing the differences in bit lengths of characters and words between the Honeywell and IBM machines, were also taken into account. Nevertheless, the bulk of the coding, being essentially ANSI-compatible FORTRAN, remained intact during the conversion process, and the errors found in this part of the code are not machine dependent.

4. DOCUMENTATION

The maintenance manual for TACWAR has been used extensively:

Command and Control Technical Center, Institute for Defense Analysis Tactical Warfare (TACWAR) Model, Computer Systems Manual Number CSM MM 237-77, Vols I, II, and III, Washington, D.C. (6 September 1977).

5. ANALYSIS BY TACWAR PROGRAM SECTION

The errors are numbered and documented point by point for each of the major sections of the TACWAR program.

Section A--Data Input and Cycle Control

Section B--Air Combat Simulation

Section C--Nuclear Combat Simulation

Section D--Ground and Air-Ground Combat Simulation and Theater Control

As mentioned previously, Supply Simulation is presently excluded from the analysis. Similarly, since the chemical combat simulation has essentially a one-to-one correspondence with the nuclear combat simulation, the chemical combat simulation has not been explicitly included. It is recommended that for every error described in the nuclear combat simulation (Section C) the corresponding location for the chemical combat simulation also be investigated for possible error.

Within each of the major areas above, the errors are documented for the main control subroutine first, then documented for the remaining subroutines as they are logically utilized in the program flow. The documentation of each error will consist of the following points.

- (1) Descriptive title,
- (2) Exact location of error (i.e., providing name of subroutine in which error occurs and portion of code that contains error),
- (3) Discussion of reasons error is believed to exist,
- (4) Proposed method to eliminate error,
- (5) Copies of applicable code before and after proposed changes.

SECTION A--DATA INPUT and CYCLE CONTROL

No errors were detected in this section.

SECTION B--AIR COMBAT SIMULATION

Error B1: Parameter NDS (IST,K) is not checked for a zero condition.

Location: Subroutine AIRMOD before DO loop 542.

Discussion: NDS (IST,K) contains the number of divisions in the active battle area of sector IST for side K. The possibility exists for NDS to be zero, in which case DO loop 542 should be skipped. This type of check for NDS = 0 occurs many places in other subroutines of TACWAR (e.g., in subroutine GC before DO loop 1010, before DO loop 2525, etc.).

Corrective Action: Place a logical GOTO statement before DO loop 542 to skip the DO loop on the condition that NDS (IST,K) is zero.

Original:

```
C      SUM OVER ALL DIVISIONS IN SECTOR IST TO OBTAIN PSRSCA(ISS,IST)
C      = NO. SHLRT RANGE SAMS ALIVE AND OPERATING OF TYPE ISS WHICH
C      DEFEND COMBAT UNITS IN SECTOR IST
      DO 542 IDS=M1,M2
C      ID = INDEX TO DIVISION LOCATION IN ACTIVE BATTLE AREA IN POSITION
C      IDS OF SECTOR IST = IDLABA(IDS,IST)
      ID=IDLABA(IDS,IST)
      DO 541 ISS=1,N3
      PSRSCA(ISS,IST)=PSRSCA(ISS,IST)+WDIV(IW+ISS,ID)
C      WDIV(IW+ISS,ID)= ACTUAL NO. TYPE(IW+ISS) WEAPONS IN DIVISION ID
      541 CONTINUE
      542 CONTINUE
```

Error B1 (Cont'd)

Corrected:

```
C   SUM OVER ALL DIVISIONS IN SECTOR IST TO OBTAIN PSRSCA(ISS,IST)
C   = NO. SHORT RANGE SAMS ALIVE AND OPERATING OF TYPE ISS WHICH
C     DEFEND COMBAT UNITS IN SECTOR IST
C     IF(NDS(K).EQ.0)GOTO 546                                I JC1001
C     DO 542 IDS=M1,M2
C     ID = INDEX TO DIVISION LOCATION IN ACTIVE BATTLE AREA IN POSITION
C     IDS OF SECTOR IST = IDLABA(IDS,IST)
C     ID=IDLABA(IDS,IST)
C     DO 541 ISS=1,N3
C     PSRSCA(ISS,IST)=PSRSCA(ISS,IST)+WDIV(IW+ISS,ID)
C     WDIV(IW+ISS,ID)= ACTUAL NO. TYPE(IW+ISS) WEAPONS IN DIVISION ID
      541 CONTINUE
      542 CONTINUE
      546 CONTINUE                                              I JC1001
```

Error B2: Incorrect use of local parameters PSRSC(ISS,IS,K),
PSRSI(ISS,IS,K), PSRSCA(ISS,IST), and PSRSIA(ISS,IST).

Location: Subroutine AIRMOD within DO loop 57010 and subroutine
AIRATT within DO loop 540.

Discussion: This is a rather complex error, spanning several subroutines and apparently involving a conflict between the use of the indices IS and IST to represent geometric regions on the one hand and sectors on the other. PSRSC(ISS,IS,K) and PSRSI(ISS,IS,K) are local parameters (within the air combat model) that contain the initial number of short-range (surface-to-air missile) SAM weapons of type ISS within divisions in the active battle area (for PSRSC), and the first inactive battle area (for PSRSI) in sector IS for side K. Similarly, PSRSCA(ISS,IST) and PSRSIA(ISS,IST) are local parameters that contain the dynamic number of operational SR-SAM weapons of type ISS within sector IST for active battle area divisions and first inactive battle area divisions, respectively. PSRSCA and PSRSIA are correctly initialized within DO loops 541 and 543 of subroutine AIRMOD, and the results are correctly transferred to parameters PSRSC and PSRSI within DO loop 561 of subroutine AIRMOD. However, PSRSCA and PSRSIA are incorrectly reset to zero within DO loop 540 of subroutine AIRATT, which is called early within DO loop 57050 of subroutine AIRMOD. Subsequent calls to subroutines AOVL1, ATTR2 (via AOVL2), and ATTR5 then use parameters PSRSCA and PSRSIA which were previously incorrectly reset to zero. Lastly, parameters PSRSC and PSRSI are incorrectly adjusted within DO loop 57010 of subroutine AIRMOD to reflect the number of weapons killed or damaged. This adjustment is incorrect on two accounts: (a) no adjustment should be made, since PSRSC and PSRSI are used later within DO loops 57064 and 57067 to apportion the operational weapons among the participating divisions and (b) even if an adjustment were required, the index IS for parameters PSRSC(ISS,IS,K) and PSRSI(ISS,IS,K) within DO loop 57010 of subroutine AIRMOD denotes a region value which is inconsistent with the sector index value by which the parameters were initialized in DO loop 561 of subroutine AIRMOD (as described above).

Corrective Action: (a) In subroutine AIRMOD remove the statements within DO loop 57010 that adjust the parameters PSRSC and PSRSI.

(b) In subroutine AIRATT remove the statements within DO loop 540 that reset parameters PSRSCA and PSRSIA to zero.

Original:

Subroutine AIRMOD

```
C -----
C      APPLY SAM KILLS AND DAMAGES TO SAM INVENTORIES
C -----
C      SUBTRACT KILLED AND DAMAGED SAMS TO OBTAIN NO.ALIVE DEFENDING
C      COMBAT UNITS(PSRSC),INTERDICTION TARGETS(PSRSI), FORWARD AIRBASES
C      (PSRSF), REAR AIR BASES(PSRSR)
C      DO 57010 ISS=1,N3
C      JCI PSRSC(ISS,IS,K)=PSRSC(ISS,IS,K)-PSRSCK(ISS)-PSRSCD(ISS)      C JC1001
C      JCI PSRSI(ISS,IS,K)=PSRSI(ISS,IS,K)-PSRSIK(ISS)-PSRSID(ISS)      C JC1001
C      PSRSF(ISS,IS,K)=PSRSF(ISS,IS,K)-PSRSFK(ISS)-PSRSFD(ISS)
C      PSRSR(ISS,IS,K)=PSRSR(ISS,IS,K)-PSRSRK(ISS)-PSRSRD(ISS)
C      DSSMPL = NO. SHORT RANGE SAMS DAMAGED AND IN MAINTENANCE POOL
C      DSSMPL(ISS,K)=DSSMPL(ISS,K)+PSRSFD(ISS)+PSRSRD(ISS)
57010 CONTINUE
```

Subroutine AIRATT

```
DO 540  ISS=1,N3
PSRSFA(ISS)= NO.ALIVE TYPE ISS SAMS (OR AAA) DEFENDING FORWARD A/B
PSRSCA(ISS,IS)=0.
PSRSFA(ISS)=PSRSF(ISS,IS,K)*TEMP1
PSRSRA(ISS)=PSRSR(ISS,IS,K)*TEMP2
PSRSZA(ISS)=PSRSZ(ISS,K)/NS*TEMP3
C INITIALIZE SAMS KILLED(K), DAMAGED(0), SUPPRESSED(1) THIS CYCLE
PSRSIA(ISS,IS)=0.
```

Corrected:

Subroutine AIRMOD

```
C -----
C      APPLY SAM KILLS AND DAMAGES TO SAM INVENTORIES
C -----
C      SUBTRACT KILLED AND DAMAGED SAMS TO OBTAIN NO.ALIVE DEFENDING
C      COMBAT UNITS(PSRSC),INTERDICTION TARGETS(PSRSI), FORWARD AIRBASES
C      (PSRSF), REAR AIR BASES(PSRSR)
C      DO 57010 ISS=1,N3
C      PSRSC(ISS,IS,K)=PSRSC(ISS,IS,K)-PSRSCK(ISS)-PSRSCD(ISS)
C      PSRSI(ISS,IS,K)=PSRSI(ISS,IS,K)-PSRSIK(ISS)-PSRSID(ISS)
C      PSRSF(ISS,IS,K)=PSRSF(ISS,IS,K)-PSRSFK(ISS)-PSRSFD(ISS)
C      PSRSR(ISS,IS,K)=PSRSR(ISS,IS,K)-PSRSRK(ISS)-PSRSRD(ISS)
C      DSSMPL = NO. SHORT RANGE SAMS DAMAGED AND IN MAINTENANCE POOL
C      DSSMPL(ISS,K)=DSSMPL(ISS,K)+PSRSFD(ISS)+PSRSRD(ISS)
57010 CONTINUE
```

Subroutine AIRATT

```
DO 540  ISS=1,N3
C      PSRSFA(ISS)= NO.ALIVE TYPE ISS SAMS (OR AAA) DEFENDING FORWARD A/B
C      JCI PSRSCA(ISS,IS)=0.                                              C JC1001
C      PSRSFA(ISS)=PSRSF(ISS,IS,K)*TEMP1
C      PSRSRA(ISS)=PSRSR(ISS,IS,K)*TEMP2
C      PSRSZA(ISS)=PSRSZ(ISS,K)/NS*TEMP3
C      JCI PSRSIA(ISS,IS)=0.
C      INITIALIZE SAMS KILLED(K), DAMAGED(0), SUPPRESSED(1) THIS CYCLE
```

Error B3: Parameter TSRSSC(ISS,K) is not accumulated.

Location: Subroutine AIRMOD within DO loop 57032.

Discussion: Parameter TSRSSC(ISS,K) should contain the total number of short-range SAM weapons of type ISS for side K that are suppressed by air combat during the current combat cycle. The adjustment of TSRSSC within DO loop 57032 is not an accumulation but contains only the number of SAM's suppressed within the region currently being addressed by the index IS (i.e., DO parameter IS of DO loop 57050).

Corrective Action: (a) Include the parameter TSRSSC(ISS,K) in the adjustment expression with DO loop 57032 of subroutine AIRMOD to form an accumulation.

(b) Reset the parameter TSRSSC(ISS,K) to zero at a location before DO loop 57050 (e.g., within DO loop 502) of subroutine AIRMOD.

Original:

```
DO 57032 ISS=1,N3
C CUMMULATIVE NUMBERS OF SHORT RANGE SAMS KILLED
C CPSSFK(ISS,K)=CPSSFK(ISS,K)+PSRSFK(ISS)
C CPSSRK(ISS,K)=CPSSRK(ISS,K)+PSRSRK(ISS)
C CPSSCK(ISS,K)=CPSSCK(ISS,K)+PSRSCK(ISS)
C CPSSIK(ISS,K)=CPSSIK(ISS,K)+PSRSIK(ISS)
C TSRSSC = TOTAL SHORT RANGE SAMS SUPPRESSED THIS CYCLE
C CSRSDM = CUMMULATIVE SHORT RANGE SAMS DAMAGED
C TSRSSC(ISS,K)=PSRSFS(ISS)+PSRSRS(ISS)+PSRSCS(ISS)+PSRSIS(ISS)
C CSRSDM(ISS,K)=CSRSDM(ISS,K)+PSRSFD(ISS)+PSRSRD(ISS)+PSRSCD(ISS)
X +PSRSID(ISS)
57032 CONTINUE
```

Corrected:

```
C INITIALIZE COMBAT RESULTS FOR SAMS IN THI. CYCLE
DO 502 ISS=1,N3
PSRSZK(ISS)=0.
PSRSZD(ISS)=0.
PSRSZS(ISS)=0.
TSRSSC(ISS,K)=0.
502 CONTINUE

DO 57032 ISS=1,N3
C CUMMULATIVE NUMBERS OF SHORT RANGE SAMS KILLED
C CPSSFK(ISS,K)=CPSSFK(ISS,K)+PSRSFK(ISS)
C CPSSRK(ISS,K)=CPSSRK(ISS,K)+PSRSRK(ISS)
C CPSSCK(ISS,K)=CPSSCK(ISS,K)+PSRSCK(ISS)
C CPSSIK(ISS,K)=CPSSIK(ISS,K)+PSRSIK(ISS)
C TSRSSC = TOTAL SHORT RANGE SAMS SUPPRESSED THIS CYCLE I JC1001
C CSRSDM = CUMMULATIVE SHORT RANGE SAMS DAMAGED
C TSRSSC(ISS,K)=PSRSFS(ISS)+PSRSRS(ISS)+PSRSCS(ISS)+PSRSIS(ISS) I JC1001
* +TSRSSC(ISS,K) I JC1001
C CSRSDM(ISS,K)=CSRSDM(ISS,K)+PSRSFD(ISS)+PSRSRD(ISS)+PSRSCD(ISS)
X +PSRSID(ISS)
57032 CONTINUE
```

Error B4: Parameter TMRSSC(IMS,K) is not accumulated.

Location: Subroutine AIRMOD within DO loop 57034.

Discussion: This error is identical to B3, except that B4 involves parameter TMRSSC(IMS,K) referring to medium-range SAM weapons of type IMS for side K.

Corrective Action: Similar to B3.

(a) Form an accumulation at the appropriate statement in DU loop 57034 of subroutine AIRMOD.

(b) Reset the parameter TMRSSC(IMS,K) to zero at a location before DO loop 57050. This will require the introduction of a new DO loop since none presently exist which step over the index IMS.

Original:

```
57032 CONTINUE
  DD 57034 1MS=1,N4
  (  TMRSSC = TOTAL MEDIUM RANGE SAMS SUPPRESSED THIS CYCLE
  (  CBMSK = CUMMULATIVE MEDIUM RANGE SAMS KILLED
  TMRSSC(IMS,K)=BMRSS(IMS)
  CBMSK(IMS,K)=CBMSK(IMS,K)+BMRSK(IMS)
```

Corrected:

```
502 CONTINUE
  DD 529 1MS=1,N4
  TMRSSC(IMS,K)=0.
529 CONTINUE
  DD 528 1LS=1,N5
  (  TMRSSC = TOTAL MEDIUM RANGE SAMS SUPPRESSED THIS CYCLE
  (  CBMSK = CUMMULATIVE MEDIUM RANGE SAMS KILLED
  (  CMRSDM = CUMMULATIVE MEDIUM RANGE SAMS DAMAGED
  CBMSK(IMS,K)=CBMSK(IMS,K)+BMRSK(IMS)
  TMRSSC(IMS,K)=BMRSS(IMS) +TMRSSC(IMS,K)           I JC1001
  CMRSDM(IMS,K)=CMRSDM(IMS,K)+BMRSD(IMS)           I JC1001
57034 CONTINUE
```

Error B5: Parameter TLRSSC(ILS,K) is not accumulated.

Location: Subroutine AIRMOD within DO loop 57036.

Discussion: This error is identical to B3 except that B5 involves parameter TLRSSC(ILS,K) referring to long-range SAM weapons of type ILS for side K.

Corrective Action: Similar to B3.

(a) Form an accumulation at the appropriate statement in DO loop 57036 of subroutine AIRMOD.

Error B5 (Cont'd)

(b) Reset the parameter TLRSSC(ILS,K) to zero at a location before DO loop 57050 (e.g., within DO loop 528 of subroutine AIRMOD).

Original:

```
57034 CONTINUE
  DD 57036 ILS=1,N5
C   CLRSDM= CUMMULATIVE LONG RANGE SAMS DAMAGED
C   TLRSSC = TOTAL LONG RANGE SAMS KILLED THIS CYCLE
C   CALSRK = CUMMULATIVE LONG RANGE SAMS KILLED
CLRSDM(ILS,K)=CLRSDM(ILS,K)+ALRSRD(ILS)
TLRSSC(ILS,K)=          ALRSRSEILS}
CALSRK(ILS,K)=CALSRK(ILS,K)+ALRSRK(ILS)
57036 CJNTINUE
```

Corrected:

```
DD 528 ILS=1,N5
ALRSZK(ILS)=0.
ALRSZD(ILS)=0.
ALRSZS(ILS)=0.
TLRSSC(ILS,K)=0.
528 CJNTINUE
```

```
57034 CJNTINUE,
  DD 57036 ILS=1,N5
C   CLRSDM= CUMMULATIVE LONG RANGE SAMS DAMAGED
C   TLRSSC = TOTAL LONG RANGE SAMS KILLED THIS CYCLE
C   CALSRK = CUMMULATIVE LONG RANGE SAMS KILLED
CLRSDM(ILS,K)=CLRSDM(ILS,K)+ALRSRD(ILS)
TLRSSC(ILS,K)= TLRSSC(ILS,K)+ ALRSRS(ILS)      I JC1001
CALSRK(ILS,K)=CALSRK(ILS,K)+ALRSRK(ILS)
57036 CJNTINUE
```

Error B6: The line accumulating parameter CMRSDM(IMS,K) has been omitted.

Location: Subroutine AIRMOD within DO loop 57034.

Discussion: Parameter CMRSDM(ISM,K) contains the number of medium-range SAM weapons of type IMS for side K that have been damaged by air combat accumulated to the present combat cycle. The statement to perform this accumulation is apparently absent.

Error B6 (Cont'd)

Corrective Action: Include the statement forming the accumulation of CMRSDM(IMS,K) (by incrementing with parameter BMRSD(IMS)) within DO loop 57034 of subroutine AIRMOD. See as an example the appropriate statement within DO loop 57032 of subroutine AIRMOD for a similar accumulation of parameter CSRSDM(ISS,K) referring to short-range SAM weapons of type ISS for side K.

Original:

```
57032 CONTINUE
  DO 57034 IMS=1,N4
C   THRSSC = TOTAL MEDIUM RANGE SAMS SUPPRESSED THIS CYCLE
C   CBMSK = CUMMULATIVE MEDIUM RANGE SAMS KILLED
    THRSSC(IMS,K)=BMRSS(IMS)
    CBMSK(IMS,K)=CBMSK(IMS,K)+BMRSK(IMS)
57034 CONTINUE
```

Corrected:

```
C   THRSSC = TOTAL MEDIUM RANGE SAMS SUPPRESSED THIS CYCLE
C   CBMSK = CUMMULATIVE MEDIUM RANGE SAMS KILLED
C   CMRSDM = CUMMULATIVE MEDIUM RANGE SAMS DAMAGED
    CBMSK(IMS,K)=CBMSK(IMS,K)+BMRSK(IMS)
    THRSSC(IMS,K)=BMRSS(IMS) +THRSSC(IMS,K)           I JC1001
    CMRSDM(IMS,K)=(CMRSDM(IMS,K)+BMRSD(IMS))        I JC1001
57034 CONTINUE
```

Error B7: Parameter NDS(IST,K) is not checked for zero condition.

Location: Subroutine AIRMOD before DO loop 57064.

Discussion: This error is identical to B1 above.

Corrective Action: Similar to B1 above.

Original:

```
C   RFCOMPUTE WDIV= NO. WEAPONS OF EACH TYPE IN DIVISION ID TO
C   REFLECT RESULTS OF CURRENT ENGAGEMENT
  IW=NW(K)-NDSAMS(K)-1
  UO 57075 IST=1,NST
  UO 57070 ISS=1,N3
  IF(PSRSC(ISS,IST,K).LE..0001) GO TO 57065
  DO 57064 IDS=M1,M2
    ID=IDLABA(IDS,IST)
C   FOR DIVISIONS IN COMBAT WDIV= NO. WEAPONS REMAINING = NO. AT START
C   OF CYCLE*(FRACTION OF SAMS DEFENDING COMBAT UNITS STILL ALIVE)
    WDIV(IW+ISS, ID)=WDIV(IW+ISS, ID)*(PSRSC(ISS,IST)/PSRSC(ISS,IST,K))
57064 CONTINUE
57065 IF(PSRSI(ISS,IST,K).LE..0001) GO TO 57070
```

Error B7 (Cont'd)

Corrected:

```
C      RECOMPUTE WDIV= NO. WEAPONS OF EACH TYPE IN DIVISION ID TO
C      REFLECT RESULTS OF CURRENT ENGAGEMENT
IW=NW(K)-NDSAMS(K)-1
DO 57075 IST=1,NST
DO 57070 ISS=1,N3
IF(NDS(K).EQ.0)GOTO 57065
IF(PSRSC(ISS,IST,K).LE..0001) GO TO 57065
DO 57064 IDS=M1,M2
ID=IDLABA(IDS,IST)
C      FOR DIVISIONS IN COMBAT WDIV= NO. WEAPONS REMAINING = NO.AT START
C      OF CYCLE*FRACTION OF SAMS DEFENDING COMBAT UNITS STILL ALIVE
WDIV(IW+ISS,ID)=WDIV(IW+ISS,IST)*(PSRSC(ISS,IST)/PSRSC(ISS,IST,K))
57064 CONTINUE
57065 IF(PSRS1(ISS,IST,K).LE..0001) GO TO 57070
```

I JC1001

Error B8: Incorrect relational operator used.

Location: Subroutine ALLOCT at statement 12045.

Discussion: The relational expression IRAC(IAC,2,L).LE.5 is incorrect since the parameter IRAC does not contain values greater than five.

Corrective Action: Change the relational operator .LE. to .LT.

Original:

```
12040 CONTINUE
GO TO 12055
12045 IF(IRAC(IAC,2,L).LE.5.OR.WTAZ.LE.SNASBA(L)) GO TO 12050
FAAARZ(IAC,IS,L)=FAABAA(IAC,L)
FAAERZ(IAC,IS,L)=FAABAE(IAC,L)
FAASRZ(IAC,IS,L)=FAABAS(IAC,L)
GO TO 12055
12050 IF(IRAC(IAC,2,L).LT.2) GO TO 12065
```

Corrected:

```
12040 CONTINUE
GO TO 12055
12045 IF(IRAC(IAC,2,L).LT.5.OR.WTAZ.LE.SNASBA(L)) GO TO 12050 C JC1001
FAAARZ(IAC,IS,L)=FAABAA(IAC,L)
FAAERZ(IAC,IS,L)=FAABAE(IAC,L)
FAASRZ(IAC,IS,L)=FAABAS(IAC,L)
GO TO 12055
12050 IF(IRAC(IAC,2,L).LT.2) GO TO 12065
```

Error B9: Parameter FACSFC(IAC,IS,I) is not reset to zero.

Location: Subroutine ALLOCT within DO loop 13005.

Discussion: FACSFC(IAC,IS,I) is one of the several parameters that contain the fractional allocation of aircraft type IAC, based in region IS of side L attacking targets in region KS of side K (where $I = KS + NNR \cdot (L-1)$, NNR = total number of regions for side L. Whereas the remaining parameters--e.g., FACAFc(IAC,IS,I), etc.--are reset to zero within DO loop 13005 of subroutine ALLOCT, parameter FACSFC(IAC,IS,I) is omitted.

Corrective Action: Provide a statement within DO loop 13005 of subroutine ALLOCT that resets the parameter FACSFC(IAC,IS,I) to zero.

Original:

```
DO 13005 KS=1,NS
I=KS+NNR*(L-1)
FACAFc(IAC,IS,I)=0.
FACEFc(IAC,IS,I)=0.
FAAAFF(IAC,IS,I)=0.
FAAAFR(IAC,IS,I)=0.
FAAEFF(IAC,IS,I)=0.
FAAEFR(IAC,IS,I)=0.
FAIAFF(IAC,IS,I)=0.
FAIEFF(IAC,IS,I)=0.
FAISFF(IAC,IS,I)=0.
FAASFR(IAC,IS,I)=0.
FAASFF(IAC,IS,I)=0.
FABSFF(IAC,IS,I)=0.
```

Corrected:

```
DO 13005 KS=1,NS
I=KS+NNR*(L-1)
FACAFc(IAC,IS,I)=0.
FACEFc(IAC,IS,I)=0.
FAAAFF(IAC,IS,I)=0.
FAAAFR(IAC,IS,I)=0.
FAAEFF(IAC,IS,I)=0.
FAAEFR(IAC,IS,I)=0.
FAIAFF(IAC,IS,I)=0.
FAIEFF(IAC,IS,I)=0.
FAISFF(IAC,IS,I)=0.
FAASFR(IAC,IS,I)=0.
FAASFF(IAC,IS,I)=0.
FACSFC(IAC,IS,I)=0.
FABSFF(IAC,IS,I)=0.
```

Error B10: Expressions do not adjust for PAA4 and PAA5.

Location: Subroutine ALLOCT following statement 13045.

Error B10 (Cont'd)

Discussion: The local parameters PAA4 and PAA5 are used to adjust the fractional allocations of aircraft from forward airbases assigned to close air support (CAS) and airbase attack and escort missions, respectively. Specifically, when aircraft from rear airbases are diverted from their primary enemy airbase targets to CAS targets because of range constraints, then an apportioned number of aircraft from forward airbases are simultaneously diverted from their primary CAS targets to enemy airbase targets, provided that range constraints permit such diversions. For example, if the range constraint IF statement at 13015 of subroutine ALLOCT is not effected, then DO loops 13020 and (possibly) 13025 are performed. Within DO loop 13025 the fractional allocation parameters FAABAA(IAC,L), and FAABAE(IAC,L) are adjusted (incremented) by parameters PAA4 and PAA5, respectively. Later, within DO loop 13060, parameters FACASA(IAC,L) and FACASE(IAC,L) are adjusted (decremented) by the same parameters PAA4 and PAA5, respectively. A similar situation occurs for the IF statement at 13030 and DO loops 13035 and 13040. However, if program flow causes control to pass through IF statement 13045, then the expected incremented adjustments to parameters FAABAA and FAABAE do not occur.

Corrective Action: Following statement 13045 of subroutine ALLOCT, increment the expressions FAABAA(IAC,L) and FAABAE(IAC,L) by PAA4 and PAA5, respectively.

Original:

```
13040 CONTINUE
GO TO 13055
13045 IF(IRAC(IAC,1,L).LT.5.OR.WTAZ.LE.SNASBA(L)) GO TO 13050
FAAFFZ(IAC,IS,L)=FAABAA(IAC,L)
FAAEFZ(IAC,IS,L)=FAABAE(IAC,L)
FAASFZ(IAC,IS,L)=FAABAS(IAC,L)
GO TO 13055
```

Corrected:

```
13040 CONTINUE
GO TO 13055
13045 IF(IRAC(IAC,1,L).LT.5.OR.WTAZ.LE.SNASBA(L)) GO TO 13050
FAAFFZ(IAC,IS,L)=FAABAA(IAC,L)+PAA4
FAAEFZ(IAC,IS,L)=FAABAE(IAC,L)+PAA5
FAASFZ(IAC,IS,L)=FAABAS(IAC,L)
GO TO 13055
C JC1001
C JC1001
```

Error B11: Parameter IRAC(IAC,1,L) is not checked for condition of less than two.

Location: Subroutine ALLOCT at statement 1^;50.

Discussion: Parameter IRAC(IAC,1,L) contains index values ranging from one to five that refer to range values of type IAC aircraft based on forward airbases for side L. Typically, the program control flows through different parts of the aircraft allocation sections, depending on the range constraints imposed by the parameter IRAC. As an example, for aircraft based on rear region airbases at statement 12050, control is passed to statement 12065 if IRAC(IAC,2,L) is less

Error B11 (Cont'd)

than two. For the corresponding situation for aircraft based on forward region airbases, as statement 13050 there is no range constraint check for IRAC(IAC,1,L) less than two.

Corrective Action: (a) Include an IF statement at 13050 in subroutine ALLOCT to transfer program control to statement 13065 (a new statement number) for the condition IRAC(IAC,1,L).LT.2. This condition implies that all aircraft of type IAC based forward for side L are by range constraint to be assigned airbase defense missions.

(b) At the new statement 13065 set the fractional allocation parameter for forward airbase defense (FAADFF(IAC,IS,L)) to unity reflecting the mission allocation requirements imposed by (a) above.

(c) Following the new statement 13065, transfer program control to statement 13080.

(d) If the coding for section 130 is to be similar to section 120 of subroutine ALLOCT, then new statement number 13070 can be placed at the beginning of the DO loop 13075, and a new statement giving unconditional transfer to 13070 can be inserted following DO loop 13060. The statements described in (b) and (c) above can then be inserted between the new unconditional transfer statement to 13070 and statement 13070.

Original:

```
13050 TEMP=0.0
IF(IRAC(IAC,1,L).LT.3) TEMP=1.0
APA1=FAABAA(IAC,L)+TEMP*FAINDA(IAC,L)
APA2=FAABAE(IAC,L)+TEMP*FAINDE(IAC,L)
APA3=FAABAS(IAC,L)+TEMP*FAABAS(IAC,L)
13055 FAADFF(IAC,IS,L)=FAABAU(IAC,L)
FACDFC(IAC,IS,L)=FACASD(IAC,L)
DO 13060 KS=1,NS
I=KS+NNR*(L-1)
FACAF(IAC,IS,I)= (FACASAI(IAC,L)+APA1-PAA4)*FCASSS(IS,KS,L)
FACEFC(IAC,IS,I)= (FACASE(IAC,L)+APA2-PAA5)*FCASSS(IS,KS,L)
FACSFC(IAC,IS,I)= FACASS(IAC,L)*FCASSS(IS,KS,L)
13060 CONTINUE
DO 13075 KS=1,NS
I=KS+NNR*(L-1)
FABSFF(IAC,IS,I)= FABASS(IIS,KS,L)*(FABSUP(IAC,L)+APA3)
13075 CONTINUE
```

Error B11 (Cont'd)

Corrected:

```
13050 IF(IRAC(IAC,1,L).LT.2)GOTO 13065          I JC1001
      TEMP=0.0
      IF(IRAC(IAC,1,L).LT.3) TEMP=1.0
      APA1=FAABAA(IAC,L)+TEMP*FAINDA(IAC,L)+PAA4
      APA2=FAABAE(IAC,L)+TEMP*FAINDE(IAC,L)+PAAS
      APA3=FAABAS(IAC,L)+TEMP*FAINDS(IAC,L)          C JC1001
                                                     C JC1001
                                                     C JC1001
13055 FAADFF(IAC,IS,L)=FAABAD(IAC,L)
      FACDFC(IAC,IS,L)=FACASD(IAC,L)
      DO 13060 KS=1,NS
      I=KS+NNR*(L-1)
      FACAF(IAC,IS,I)=(FACASA(IAC,L)+APA1-PAA4)*FCASSS(IS,KS,L)
      FACEFC(IAC,IS,I)=(FACASE(IAC,L)+APA2-PAA5)*FCASSS(IS,KS,L)
      FACSFC(IAC,IS,I)=FACASS(IAC,L)*FCASSS(IS,KS,L)
13060 CONTINUE
      GOTO 13070
13065 FAADFF(IAC,IS,L)=1.          I JC1001
      GOTU 13080
13070 DO 13075 KS=1,NS          I JC1001
      I=KS+NNR*(L-1)
      FABSFF(IAC,IS,I)=FABASS(IS,KS,L)*(FABSUP(IAC,L)+APA3)          C JC1001
13075 CONTINUE
C
13080 CONTINUE
13085 CONTINUE
```

Error B12: Parameter FAINDS(IAC,L) omitted.

Location: Subroutine ALLOCT before statement 13055.

Discussion: The expression for local variable APA3 contains the term TEMP*FAABAS(IAC,L). APA3 specifies the adjustment to both SAM suppression allocations diverted from airbase and interdiction target suppression allocations because of range constraints (or "unworthy" target constraints). Thus, the term in question should reflect the adjustments in allocation diverted from interdiction targets (i.e., TEMP*FAINDS(IAC,L)).

Corrective Action: Change the parameter FAABAS(IAC,L) to the parameter FAINDS(IAC,L) in the affected statement.

Original:

```
13.50 TEMP=0.0
      IF(IRAC(IAC,1,L).LT.3) TEMP=1.0
      APA1=FAABAA(IAC,L)+TEMP*FAINDA(IAC,L)
      APA2=FAABAE(IAC,L)+TEMP*FAINDE(IAC,L)
      APA3=FAABAS(IAC,L)+TEMP*FAABAS(IAC,L)
13055 FAADFF(IAC,IS,L)=FAABAD(IAC,L)
      FACDFC(IAC,IS,L)=FACASD(IAC,L)
```

Error B12 (Cont'd)

Corrected:

```
13050 IF(IRAC(IAC,1,L).LT.2)GOTO 13065           I JC1001,  
        TEMP=0.0  
        IF(IRAC(IAC,1,L).LT.3) TEMP=1.0  
        APA1=FAAEAA(IAC,L)+TEMP*FAINDA(IAC,L)+PAA4  
        APA2=FAABAE(IAC,L)+TEMP*FAINDE(IAC,L)+PAA5  
        APA3=FAABAS(IAC,L)+TEMP*FAINDS(IAC,L)  
13055 FAADFF(IAC,IS,L)=FAABAD(IAC,L)  
        FACDFC(IAC,IS,L)=FACASD(IAC,L)             C JC1001  
                                                C JC1001  
                                                C JC1001
```

Error B13: Questionable use of air combat interaction parameter CA.

Location: Subroutine AOVLL in call to subroutine ATRTDA following DO loop 6020.

Discussion: The formal parameter CA in subroutine ATRTDA represents the number of independent air combat engagements that can occur between two combative elements being analyzed by ATRTDA. The actual parameter used in the CALL statement following DO loop 6020 of subroutine AOVLL is CA, which was initialized to the value WIDS(IS)/WCACAS (or unity, whichever is larger) at the beginning of subroutine ATTR2. This value of CA represents the number of independent engagements between CAS aircraft defending across the region indexed IS for side K against enemy (side L) aircraft penetrating uniformly across the region. However, the CALL statement to ATRTDA in question is used to determine the air combat attrition of CAS defenders against airbase target and interdiction target attack and suppressor aircraft that are not penetrating uniformly across the region but only in a number of corridors given by the parameter PENCOR(IS,L). Thus, the number of independent combat engagements is parameter PENCOR, not CA; this situation is similar to the earlier statement, which calls subroutine ATRTED following DO loop 5010, in which the parameter PENCOR is, in fact, used.

Corrective Action: In the affected CALL statement, substitute the parameter PENCOR(IS,L) for the parameter CA as the actual parameter corresponding to the CA formal parameters of the subroutine ATRTDA.

Error B13 (Cont'd)

Original:

```

6020 CONTINUE
C
C      CALCULATE ATTRITION OF DEFENDERS (D),ATTACKERS(A),SUPPRESSORS(S)
CALL ATRIDA(ID,TA,TS,PDCDFB(1,K),TLMH,SUMM,SUMM1,PAJOR(1,L),
X      PSJDR(1,L),TUMMI,FKLAA(1,L),FKLAA(1,K),AEDGE(1,K),CA,N1,N2,
X      DA,DK,DH,DD,AA,AK,AH,AD,SA,SK,SH,SD)
DU 6030 KAC=1,N2
CAKAS(1,KAC,L)=CAKAS(1,KAC,L) + (DD(KAC) + DK(KAC))/1
      AMA(1(1.0,SRACH(KAC,2,K))

```

Corrected:

```

6020 CONTINUE
C
C      CALCULATE ATTRITION OF DEFENDERS (D),ATTACKERS(A),SUPPRESSORS(S)
CALL ATRIDA(ID,TA,TS,PDCDFB(1,K),TUMM,SUMM,SUMM1,PAJOR(1,L),
X      PSJOK(1,L),TUMM1,FKLAA(1,L),FKLAA(1,K),AEDGE(1,K),PENCORIS,L) C JC1001
X      ,N1,N2,DA,DK,DH,DD,AA,AK,AH,AD,SA,SK,SH,SD) C JC1001
DU 6030 KAC=1,N2
CAKAS(1,KAC,L)=CAKAS(1,KAC,L) + (DD(KAC) + DK(KAC))/1
      AMA(1(1.0,SRACH(KAC,2,K))

```

Error B14: Incorrect use of region index IS for section index IST.

Location: Subroutine ATTR2 in CALL statement to subroutine ATSPSS following DO loop 21010.

Discussion: The CALL statement to subroutine ATSPSS in question contains the actual parameters TAINTD(IS,K) where IS is the current region index for side L air attacks. However, the parameter TAINTD(IST,K) was previously initialized at the beginning of DO loop 24010 in subroutine ATTR2, where the index IST refers to the sectors contained in region IS.

Corrective Action: In the CALL statement to subroutine ATSPSS in question, change the actual parameter TAINTD(IS,K) to :read TAINTD(IST,K).

Original:

```

C      CALCULATE INTERDICTION SUPPRESSION A/C VS SAM (PSRSIA) ATTRITION
CALL ATSPSS(PSRSIA(1,IST),INTDSA,FSSSMS(L),PDSSPS(L),PSSSSS(1,L),
X      SUMM1,FKLASM(K),AM7RS(1,K),PDPSSS(1,K),PKSSA(1,K),
X      FKLSMA(K),TAINTD(1S,K),N3,N1,SA,SS,SK,SD,AA,AK,AD)
C      SUPPRESSION SORTIES ALIVE(A),DAMAGED(D),KILLED(K) AFTER ENGAGEMENT

```

Corrected:

```

21010 CONTINUE
C
C      CALCULATE INTERDICTION SUPPRESSION A/C VS SAM (PSRSIA) ATTRITION
CALL ATSPSS(PSRSIA(1,IST),INTDSA,FSSSMS(L),PDSSPS(L),PSSSSS(1,L),
X      SUMM1,FKLASHM(K),AM7RS(1,K),PDPSSS(1,K),PKSSA(1,K),
X      FKLSMA(K),TAINTD(1ST,K),N3,N1,SA,SS,SK,SD,AA,AK,AD) C JC1001
C      SUPPRESSION SORTIES ALIVE(A),DAMAGED(D),KILLED(K) AFTER ENGAGEMENT

```

Error B15: Incorrect use of region IS for sector index IST.

Location: Subroutine ATTR2 in CALL statement to subroutine ATRTSA following DO loop 23010.

Discussion: See B14 above.

Corrective Action: Similar to B14 above.

Original:

```
23010 CONTINUE
C
C      CALCULATE INTERDICTION ATTACK VS SAM ATTRITION.
C      THE PROB. OF ABORTING UNDAMAGED =C.O, AVG. NO. SHOTS PER SAM
C      FIRE CONTROL CENTER = 1.0
CALL ATRTSA(PSRSIA(1,IST),TA,PDPSSFB(1,K),PKSSA(1,K),0.0,FKLSMA(K),
X          AMTSRS(1,K),1.0,TAINTD(1,IS,K),N3,N1,AA,AK,AH,AD)
DO 23025 IAC=1,N1
```

Corrected:

```
23010 CONTINUE
C
C      CALCULATE INTERDICTION ATTACK VS SAM ATTRITION.
C      THE PROB. OF ABORTING UNDAMAGED =0.0, AVG. NO. SHOTS PER SAM
C      FIRE CONTROL CENTER = 1.0
CALL ATRTSA(PSRSIA(1,IST),TA,PDPSSFB(1,K),PKSSA(1,K),0.0,FKLSMA(K),
X          AMTSRS(1,K),1.0,TAINTD(1,IST,K),N3,N1,AA,AK,AH,AD)      C JC1001
DO 23025 IAC=1,N1
```

Error B16: Adjustment to parameter ABASZA(IAC) omitted.

Location: Subroutine ATTR3 following statement 26020.

Discussion: Following the calculated attrition from air combat between aircraft defending rear airbases for side K and aircraft penetrating the rear air bases for side L, all appropriate parameters were adjusted (e.g., ABAARA(IAC), etc.) except ABASZA(IAC).

Corrective Action: Insert the appropriate line adjusting the parameter ABASZA(IAC).

Original:

```
26020 IF(TS(IAC).EQ.0.0) GO TO 26025
TEMP=ABASRA(IAC)/TS(IAC)
ABASRA(IAC)=TEMP*SA(IAC)
ABASRD(IAC)=ABASRD(IAC)+TEMP*SD(IAC)
ABASKK(IAC)=ABASKK(IAC)+TEMP*SK(IAC)
ABASRH(IAC)=ABASRH(IAC)+TEMP*SH(IAC)
TEMP=ABASZA(IAC)/TS(IAC)
ABASZD(IAC)=ABASZD(IAC)+TEMP*SD(IAC)
ABASZK(IAC)=ABASZK(IAC)+TEMP*SK(IAC)
ABASZH(IAC)=ABASZH(IAC)+TEMP*SH(IAC)
26025 CONTINUE
```

Error B16 (Cont'd)

Corrected:

```
26020 IF(TS(IAC).EQ.0.0) GO TO 26025
      TEMP=ABASRA(IAC)/TS(IAC)
      ABASRA(IAC)=TEMP*SA(IAC)
      ABASRD(IAC)=ABASRD(IAC)+TEMP*SD(IAC)
      ABASRK(IAC)=ABASRK(IAC)+TEMP*SK(IAC)
      ABASRH(IAC)=ABASRH(IAC)+TEMP*SH(IAC)
      TEMP=ABASZA(IAC)/TS(IAC)
      ABASZA(IAC)=TEMP*SA(IAC)
      ABASZD(IAC)=ABASZD(IAC)+TEMP*SD(IAC)
      ABASZK(IAC)=ABASZK(IAC)+TEMP*SK(IAC)
      ABASZH(IAC)=ABASZH(IAC)+TEMP*SH(IAC)
      I JCI001
26025 CONTINUE
```

Error B17: Misuse of parameter FKLSAM(K).

Location: Subroutine ATTR3 following DO loop 29010 in call to subroutine ATRTSA.

Discussion: The formal parameter FKLSA of subroutine ATRTSA represents the fraction of hits that are lethal when a SAM shoots at an aircraft. The actual parameter used in the call to ATRTSA following DO loop 29010 in subroutine ATTR3 is FKLSAM(K), which represents the fraction of hits that are lethal when an enemy aircraft shoots at a SAM weapon on side K. The correct parameter to be used should be FKLSMA(K), which represents the fraction of hits that are lethal when a SAM weapon of side L shoots at an enemy aircraft.

Corrective Action: Replace FKLSAM(K) with FKLSMA(K) at the appropriate place in the subroutine call statement.

Original:

```
C -----
C      DD 29010 IAC=1,N1
C      TA = REAR ATTACK + COMMZ ATTACK + COMMZ ESCORT SORTIES
C          TA(IAC)=ABAARA(IAC)+ABAaza(IAC)+ABAEZA(IAC)
29010 CONTINUE
C
C      CALCULATE AIRCRAFT VS BELT SAM ATTRITION
C      CALL ATRTSA(ALRSRA,TA,PDASFB(K),PKLSA(K),PARHLS(K),FKLSAM(K),
C                  X          AMTLRS(1,K),ANSLR(SK),PENCOR(S,L),N5,N1,AA,AK,AH,AD)
C      USE ATRTSA RESULTS TO CALCULATE A/C SORTIES (A),(D),(K),(H)
```

Corrected:

```
C -----
C      DD 29010 IAC=1,N1
C      TA = REAR ATTACK + COMMZ ATTACK + COMMZ ESCORT SORTIES
C          TA(IAC)=ABAARA(IAC)+ABAaza(IAC)+ABAEZA(IAC)
29010 CONTINUE
C
C      CALCULATE AIRCRAFT VS BELT SAM ATTRITION
C      CALL ATRTSA(ALRSRA,TA,PDASFB(K),PKLSA(K),PARHLS(K),FKLSMA(K), C JCI001
C                  X          AMTLRS(1,K),ANSLR(SK),PENCOR(S,L),N5,N1,AA,AK,AH,AD)
C      USE ATRTSA RESULTS TO CALCULATE A/C SORTIES (A),(D),(K),(H)
```

Error B18: Misuse of parameter FKLASM(K).

Location: Subroutine ATTR4 following DO loop 38099 in call to subroutine ATRTSA.

Discussion: Same as error B17.

Corrective Action: Same as to error B17.

Original:

```
C -----
C      CALCULATE ATTACK A/C VS COMMZ BELT SAM ATTRITION
C      CALL ATRTSA(ALRSZA,ABAaza,PDASFB(K),PKLSA(K),PARHLS(K),FKLASM(K),
C                  X          AMTLRS(1,K),ANSLRS(K),PENCOR{IS,L},N5,N1,AA,AK,AH,AD)
C      USE ATRTSA RESULTS TO UPDATE ATTACK SORTIE ARRAYS
```

Corrected:

```
38099 CONTINUE
```

```
C -----
C(4000) AIRBASE ATTACK SORTIES{ABAaz} VS. BELT SAMS{ALRSZ} IN COMMZ
```

```
C -----
```

```
C      CALCULATE ATTACK A/C VS COMMZ BELT SAM ATTRITION
C      CALL ATRTSA(ALRSZA,ABAaza,PDASFB(K),PKLSA(K),PARHLS(K),FKLASM(K), C JC1001
C                  X          AMTLRS(1,K),ANSLRS(K),PENCOR{IS,L},N5,N1,AA,AK,AH,AD)
C      USE ATRTSA RESULTS TO UPDATE ATTACK SORTIE ARRAYS
```

Error B19: Incorrect determination of independent engagement areas.

Location: In subroutine ATTR4 assignment statement for parameter CA following label 40099.

Discussion: The parameter CA contains the number of independent engagement areas and is used as an actual parameter in the subsequent call to subroutine ATSPSS. This number of independent engagement areas for the present engagement should be a minimum of (a) the number of airbase raids to COMMZ area immediately behind the enemy region currently being evaluated and (b) the number of COMMZ airbases immediately behind that region. However, the minimum is determined using the parameter ABASEZ(K), which contains the total number of COMMZ airbases.

Corrective Action: Divide ABASEZ(K) by NS where NS contains the total number of enemy regions. This divides the total number of COMMZ airbases behind the different enemy regions on an equal apportionment.

Original:

```
40099 CONTINUE
```

```
C -----
C(4300) SUPPRESSION SORTIES{ABASZ} VS COMMZ POINT DEFENSES {PSRSZ}
C -----
C      CA = NO. ACTUAL AIRBASES ATTACKED IN COMMZ
C      CA=AMINI{UBABAz,ABASEZ(K)}
```

Error B19 (Cont'd)

Corrected:

40099 CONTINUE

C -----
C(4300) SUPPRESSION SURTIES(ABASZ) VS COMMZ POINT DEFENSES (PSRSZ)

C -----
C CA = NO. ACTUAL AIRBASES ATTACKED IN COMMZ
CA=AMIN1(UBABAZ,ABASEZ(K)/NS) C JC1001

Error B20: Incorrect update for parameter PSRSZK(ISS).

Location: In subroutine ATTR4 within DO loop 43025.

Discussion: The update assignment statement for parameter PSRSZK(ISS) contains the parameter PSRSZD(ISS) on the right-hand side.

Corrective Action: Change PSRSZD(ISS) to PSRSZK(ISS) on the right-hand side of the assignment statement.

Original:

43020 CONTINUE

DO 43025 ISS=1,N3
PSRSZA(ISS)=SA(ISS)
PSRSZD(ISS)=SD(ISS)+PSRSZD(ISS)
PSRSZK(ISS)=SK(ISS)+PSRSZD(ISS)
PSRSZS(ISS)=SS(ISS)+PSRSZS(ISS)

43025 CONTINUE

Corrected:

43020 CONTINUE

DO 43025 ISS=1,N3
PSRSZA(ISS)=SA(ISS)
PSRSZD(ISS)=SD(ISS)+PSRSZD(ISS)
PSRSZK(ISS)=SK(ISS)+PSRSZK(ISS)
PSRSZS(ISS)=SS(ISS)+PSRSZS(ISS)

43025 CONTINUE

Error B21: Incorrect use of independent engagement areas.

Location: In subroutine ATTR5 within call to subroutine ATSPSS following DO loop 50010.

Discussion: The actual parameter CA in the call to subroutine ATSPSS should contain the number of independent engagement areas for active division point SAMs and SAM suppression aircraft. The previous assignment statement for CA, however, determined the number of independent engagement areas for CAS attack and defense aircraft. The value assigned to CA for active division point SAMs and SAM suppression aircraft should be similar to that assigned for rear division point SAMs and SAM suppression aircraft, viz., the number of divisions involved in the engagement.

Corrective Action: The actual parameter in the call to subroutine ATSPSS should be changed from CA to NDS(IST,K) where NDS(IST,K) contains the number of active divisions in sector IST for side K.

Error B21 (Cont'd)

Original:

```
50010 CONTINUE
C
C      CALCULATE SUPPRESSOR (CASS) VS. POINT SAM (PSRSC) ATTRITION
CALL ATSPSS(PSRSCA(1,IST),CASSA,FSSSMS(L),PDSSPS(L),PSSSSS(1,L),
X           SUMM1,FKLASHM(K),AMTSRS(1,K),PDPSSS(1,K),PKSSA(1,K),
X           FKLMSA(K),CA,N3,N1,SA,SS,SK,SD,AA,AK,AD)
C      USE ATSPSS RESULTS TO SET SUPPRESSOR ARRAYS
```

Corrected:

```
50010 CONTINUE
C
C      CALCULATE SUPPRESSOR (CASS) VS. POINT SAM (PSRSC) ATTRITION
C JCI
CALL ATSPSS(PSRSCA(1,IST),CASSA,FSSSMS(L),PDSSPS(L),PSSSSS(1,L),
X           SUMM1,FKLASHM(K),AMTSRS(1,K),PDPSSS(1,K),PKSSA(1,K),
X           FKLMSA(K),TACASD(IST,K),N3,N1,SA,SS,SK,SD,AA,AK,AD) I JCI001
C JCI
C      USE ATSPSS RESULTS TO SET SUPPRESSOR ARRAYS
```

Error B22: Incorrect actual parameter used in calls to BINOAT.

Location: In subroutine ATRTSS following DO loop 12.

Discussion: The three calls to subroutine BINOAT should be returning values for parameters SK, SD, and SS, respectively. Instead, values for the single parameter SK are returned in each case.

Corrective Action: For the last two calls to subroutine BINOAT change the actual parameters from SK to SD and SS, respectively.

Original:

```
10 CONTINUE
CALL BINCAT(A,S,PDA,PK,CA,N1,NX,SK)
DO 12 IAC=1,N1
  PD(IAC)=PDA
12 CONTINUE
C
C      CALCULATE SK = NO. SAMS KILLED (LETHAL)
CALL BINCAT(A,S,PD,PK,CA,N1,NX,SK)
C
C      CALCULATE SD = NO. SAMS KILLED (LETHAL + NONLETHAL)
CALL BINCAT(A,S,PD,PKK,CA,N1,NX,SK)
C
C      CALCULATE SS = TOTAL NO. SAMS SUPPRESSED (INCLUDES TOTAL KILLS)
CALL BINCAT(A,S,PD,PKKK,CA,N1,NX,SK)
```

Error B22 (Cont'd)

Corrected:

```
10 CONTINUE
  CALL BINUAT(A,S,PDA,PK,CA,N1,NX,SK)
  DO 12 IAC=1,N1
  PDI(IAC)=PDA
12 CONTINUE
C
C   CALCULATE SK = NO. SAMS KILLED (LETHAL)
  CALL BINUAT(A,S,PD,PK,CA,N1,NX,SK)
C
C   CALCULATE SD = NO. SAMS KILLED (LETHAL + NONLETHAL)
  CALL BINDAT(A,S,PD,PKK,CA,N1,NX,SD)          C JCI001
C
C   CALCULATE SS = TOTAL NO. SAMS SUPPRESSED (INCLUDES TOTAL KILLS)
  CALL BINOAT(A,S,PD,PKKK,CA,N1,NX,SS)          C JCI001
```

Error B23: Parameter REQSC incorrectly reset.

Location: In subroutine DEG within DO loop 98.

Discussion: Local parameter REQSC is used to hold the total amount of airbase supplies required by COMMZ airbases for the current combat cycle. This parameter is reset within DO loop 98 for each combat sector IS and is set only on the condition IS equals 1 (i.e., on the first pass through the DO loop). Consequently, REQSF is incorrectly reset to zero when the DO loop is complete.

Corrective Action: Move the reset assignment statements for REQSC to a location above the beginning of DO loop 98.

Original:

```
DD99L=1,2
KT=NAC(L)
M=3-L
C   CI = FRACTION OF BASE OPERATING CAPABILITY DESTROYED BY EACH
C       ATTACKING AIRCRAFT
  CI=CLPABA(L)
  ZAC=0.
C   CALCULATE DEGRADATIONS FOR EACH SECTOR IS
  DD98 IS=1,NS
  REQSF=0.
  REQSR=0.
  ZAF=0.
  ZAR=0.
  REQSC=0.
C   REQSF,R = SUPPLY CONSUMPTION (TONS/DAY) =(NO.A/C)*CONSUMPTION RATE
  DO3I=1,KT
  REQSF=REQSF+ACFST(I,IS,L)*CRSAC(I,L)
  REQSR=REQSR+ACRST(I,IS,L)*CRSAC(I,L)
  IF (IS.EQ.1)REQSC=REQSC+ACCZ(I,L)*CRSAC(I,L)
C   ZAC,F,R = TOTAL NO. SUCCESSFUL ABA SORTIES BY ENEMY LAST CYCLE
  IF (IS.EQ.1)ZAC=ZAC+SABAZ(I,M)
  ZAF=ZAF+SABAFL(I,IS,M)
  3 ZAR=ZAR+SABAR(I,IS,M)
  SDF=1.
  SDR=1.
```

Error B23 (Cont'd)

Original (Cont'd)

```
C SDF,SDR =FRACTION OF SORTIES THAT CAN BE SUPPLIED =SUPPLIES/REQMT.  
IF(SABFS{IS,L}<.LT.REQSF)SDF=SABFS{IS,L}/REQSF  
IF(SABRS{IS,L}<.LT.REQSR)SDR=SABRS{IS,L}/REQSR  
C DEGSRF,R = DEGRADATION IN AIRBASE SORTIE RATE CAPABILITY  
C = MIN OF(FRAC. THAN CAN BE SUPPLIED,FRACTION NOT  
C DESTROYED BY ENEMY AIRBASE ATTACKS)  
DEGSRF{IS,L}=AMIN1(SDF,DEGSRF{IS,L}*EXP(-ZAF*C1))  
DEGSRR{IS,L}=AMIN1(SDR,DEGSRR{IS,L}*EXP(-ZAR*C1))  
C SABFS,R = TONS OF SUPPLIES REMAINING AFTER CURRENT CYCLE  
SABFS{IS,L}=AMAX1(0.,SABFS{IS,L}-REQSF)  
SABRS{IS,L}=AMAX1(0.,SABRS{IS,L}-REQSR)  
98 CONTINUE  
SDC=1.0  
C COMMZ SUPPLIES REMAINING AND DEGRADATION  
IF(SABCZ{L}<.LT.REQSC)SDC=SABCZ{L}/REQSC  
SABCZ{L}=AMAX1(0.,SABCZ{L}-REQSC)  
DEGSRC{L}=AMIN1(SDC,DEGSRC{L}*EXP(-ZAC*C1))  
99 CONTINUE
```

Corrected:

```
KT=NAC(L)  
M=3-L  
KTT=NAC(M)  
C C1 =FRACTION OF BASE OPERATING CAPABILITY DESTROYED BY EACH  
ATTACKING AIRCRAFT  
C1=CLPABA(L)  
ZAC=0.  
REQSC=0.  
C CALCULATE DEGRADATIONS FOR EACH SECTOR IS  
DO 98 IS=1,NS  
REQSF=0.  
REQSR=0.  
ZAF=0.  
ZAR=0.  
C REQSF,R = SUPPLY CONSUMPTION (TONS/DAY) =INU.A/C)*CONSUMPTION RATE  
DO 3 I=1,KT  
REQSF=REQSF+ACFST{I,IS,L}*CRSAC{I,L}  
REQSR=REQSR+ACRST{I,IS,L}*CRSAC{I,L}  
IF(IS.EQ.1)REQSC=REQSC+ACCZ{I,L}*CRSAC{I,L}  
3 CONTINUE  
DO 2 I=1,KTT  
C ZAC,F,R = TOTAL NO. SUCCESSFUL ABA SORTIES BY FNEHY LAST CYCLE  
IF(IS.EQ.1)ZAC=ZAC+SABA{I,M}  
ZAF=ZAF+SAPAF{I,IS,M}  
ZAR=ZAR+SABAR{I,IS,M}  
2 CONTINUE  
SDF=1.  
SDR=1.  
C SDF,SDR =FRACTION OF SORTIES THAT CAN BE SUPPLIED =SUPPLIES/REQMT.  
IF(SABFS{IS,L}<.LT.REQSF)SDF=SABFS{IS,L}/REQSF  
IF(SABRS{IS,L}<.LT.REQSR)SDR=SABRS{IS,L}/REQSR  
C DEGSRF,R = DEGRADATION IN AIRBASE SORTIE RATE CAPABILITY  
C = MIN OF(FRAC. THAN CAN BE SUPPLIED,FRACTION NOT  
DESTROYED BY ENEMY AIRBASE ATTACKS)
```

C JC1001

M JC1001

I JC1001

M JC1001

M JC1001

M JC1001

I JC1001

Error B23 (Cont'd)

Corrected (Cont'd)

```
C DEGSRF{IS,L}=AMIN1(SDF,DEGSRF{IS,L})*EXP(-ZAF*C1) C JC1001
C DEGSRR{IS,L}=AMIN1(SDR,DEGSRR{IS,L})*EXP(-ZAR*C1) C JC1001
C SABFS,R = TONS OF SUPPLIES REMAINING AFTER CURRENT CYCLE
C SABFS{IS,L}=AMAX1(0.,SABFS{IS,L}-REQSF)
C SABRS{IS,L}=AMAX1(0.,SABRS{IS,L}-REQSR)
98 CONTINUE
SDC=1.0
C COMMZ SUPPLIES REMAINING AND DEGRADATION
IF(SABCZ{L}.LT.REQSC)SDC=SABCZ{L}/REQSC
SABCZ{L}=AMAX1(0.,SABCZ{L}-REQSC)
DEGSRC{L}=AMIN1(SDC,DEGSRC{L})*EXP(-ZAC*C1) C JC1001
99 CONTINUE
```

SECTION C - NUCLEAR COMBAT SIMULATION

Error C1: Incorrect index used for combatant side.

Location: In subroutine NUC3 in assignment statement for parameter MAXNPT.

Discussion: In the assignment statement for MAXNPT, the parametric factor FRMAX(ISU,JE,L) is used on the right-hand side. However, in the context of the DO loop 200, L refers to the index of the combatant side being targetted while the parameter FRMAX(ISU,JE,L) contains the maximum fraction to type ISU subunits that side L may target when in escalation state JE.

Corrective Action: Use the index K (representing the targetting combatant side) in the parameter FRMAX, viz., FRMAX (ISU,JE,K).

Original:

```
C DETERMINE MAXIMUM NUMBER OF SUBUNITS ALLOWED TO BE TARGETED
C
C MAXNPT=NSUTD{ISU,1D}*FRMAX{ISU,JE,L}+0.5
C
C IF(NPT{ISU,IZ,IDS}+NPTAS{ISU,IDS}>.GT.MAXNPT) GO TO 10
NPTAS{ISU,IDS}=NPTAS{ISU,IDS}+NPT{ISU,IZ,IDS}
GO TO 50
```

Corrected:

```
C DETERMINE MAXIMUM NUMBER OF SUBUNITS ALLOWED TO BE TARGETED
C
C MAXNPT=NSUTD{ISU,1D}*FRMAX{ISU,JE,K}+0.5 C JC1001
C
C IF(NPT{ISU,IZ,IDS}+NPTAS{ISU,IDS}>.GT.MAXNPT) GO TO 10
NPTAS{ISU,IDS}=NPTAS{ISU,IDS}+NPT{ISU,IZ,IDS}
GO TO 50
```

Error C2: Incompatible scaling of civilian population centers.

Location: In subroutine NUC4 in data statement for parameter POPLM and in data initialization file for parameter PDMMX(JE,L).

Error C2 (Cont'd)

Discussion: In the data statement for parameter POPLM the data correspond to population center thresholds expressed in thousands of people (i.e., POPLM(1) = 5.0 corresponds to 5000 people), whereas the data initialization for parameter PDMMX(JE,L) contains the escalation state constraint population expressed in people (i.e., DPMMX(1,1) = 3000, corresponding to 3000 people). Since these two parameters are compared in subroutine NUC4 (within DO loop 100), they should be scaled the same.

Corrective Action: Scale the parameter POPLM data in the data statement to reflect population center thresholds expressed in people instead of thousands of people.

Original:

```
C PERCENTAGE OF ZONE POPULATIONS HELD IN  
C CITY POPULATIONS OF      '5000   '10K   '25K   '50K   '100K  
C  
C DATA(POPLM(L),L=1,5)/$0.1,0.0,25.0,50.0,100.0/  
C
```

Corrected:

```
C PERCENTAGE OF ZONE POPULATIONS HELD IN  
C CITY POPULATIONS OF      '5000   '10K   '25K   '50K   '100K  
C  
C DATA POPLM/5000.,10000.,25000.,50000.,100000./  
C
```

Error C3: Incorrect use of index for combatant side.

Location: In subroutine NUC4

- in assignment statement for parameter JE
- in parameter PDMMX at four locations.

Discussion: In the context of DO loop 600, index L represents the combatant side being targetted. However, in the assignment statement for parameter JE, the parameter IESC(IS,ITC,L) is used, representing the escalation state for targetting side L in sector IS against target type ITC. Similarly, the parameter PDMMX(JE,L) representing the minimum population of a city which constitutes a collateral damage constraint to targetting side L in escalation state JE, is also an inconsistent use of the index L.

Corrective Action: For the parameters IESC and PDMMX, substitute the index K representing the targetting combatant side for the index L.

Original:

```
C -----  
C WHAT IS THE CURRENT LEVEL OF ESCALATION  
C -----  
C  
C JE=IESC(IS,ITC,L)  
C -----  
C WHICH BATTLE AREA IS THE ACTIVE BATTLE AREA  
C -----  
C JBA=IABAS(IIS)
```

Error C3 (Cont'd)

Original (Cont'd)

```
C IF(INDS(IS,L).EQ.0) GO TO 600
C FIND INDEX TO CITY SIZE USED FOR TARGETING CONSTRAINT
C DO 100 J=1,5
KJ=J
C IF(PDMMX(JE,L).LE.POPLM(J)) GO TO 200
C 100 CONTINUE
C CITY SIZE FOR CONSTRAINT IS GREATER THAN LARGEST IN DATA BANK
C ZOLIM = PZPI(JBA,5)*POPLM(5)/PDMMX(JE,L)
C .
C GO TO 500
C IS THE CITY SIZE LESS THAN SMALLEST ONE
C 200 CONTINUE
C IF(KJ.EQ.1)GO TO 300
C CITY SIZE IS BETWEEN INDEX 1 AN 5
C KJB=KJ-1
C ZOLIM = PZPI(JBA,KJB) + (PZPI(JBA,KJ) - PZPI(JBA,KJB))
1 *(PDMMX(JE,L)- POPLM(KJB))/(POPLM(KJ) - POPLM(KJB))
C GO TO 500
C LESS THAN SMALLEST CITY
C 300 CONTINUE
ZOLIM=PZPI(JBA,1)+(POPLM(1)-PDMMX(JE,L))/POPLM(1)*61.-PZPI(JBA,1))
```

Corrected:

```
C WHAT IS THE CURRENT LEVEL OF ESCALATION
C -----
C JE=IESC(1$,$ITC,$K)
C                                     C JC1001
C -----
C WHICH BATTLE AREA IS THE ACTIVE BATTLE AREA
C -----
C JBA=TABAS($IS)
C
C IF(INDS($IS,$L).EQ.0) GO TO 600
C FIND INDEX TO CITY SIZE USED FOR TARGETING CONSTRAINT
```

Error C3 (Cont'd)

Corrected (Cont'd)

```
C      DD 100 J=1,5  
C      KJ=J  
C      IF(PDMMX(JE,K).LE.POPLM(J)) GO TU 200          C JCI001  
C      100 CONTINUE  
C      CITY SIZE FOR CONSTRAINT IS GREATER THAN LARGEST IN DATA BANK  
C      ZOLIM = PZPI(5,JBA)*POPLM(5)/PDMMX(JE,K)          C JCI001  
C      GO TO 500  
C      IS THE CITY SIZE LESS THAN SMALLEST ONE  
C      200 CONTINUE  
C      IF(KJ.EQ.1)GO TO 300  
C      CITY SIZE IS BETWEEN INDEX 1 AND 5  
C      KJB=KJ-1  
C      ZOLIM = PZPI(KJB,JBA) + (PZPI(KJ,JBA) - PZPI(KJB,JBA))  
1 *(PDMMX(JE,K)- POPLM(KJB))/(POPLM(KJ) - POPLM(KJB))          C JCI001  
C      GO TO 500  
C      LESS THAN SMALLEST CITY  
C      300 CONTINUE  
ZOLIM=PZPI(1,JBA)*(POPLM(1)-PDMMX(JE,K))/POPLM(1)*(1.-PZPI(1,JBA))C JCI001  
C
```

Error C4: Incorrect accumulation of warhead usage.

Location: In subroutine NUC5 in assignment accumulation statements for parameters KWHDW, KWHSW, and KWHTW (following labels 510, 520, and 530, respectively).

Discussion: The three parameters KWHDW(LIWS,IYL,IS), KWHSW(LIWS,IYL,IS), and KWHTW(LIWS,IYL,IS) contain the accumulated number of nuclear warheads of yield index IYL delivered by the side L type of IWS system in sector IS for divisional, sector, and theater delivery systems, respectively. However, following labels 510, 520, and 530, respectively, these parameters are incremented by unity regardless of the actual number of warheads allocated to current targets on the target list. Specifically, multiple targets (and corresponding allocation of multiple warheads) may exist for division subunits and SSM sites. The actual number of warheads allocated to each target type on the target assignment list is contained in parameter IWLBT (except for airbase targets, in which case the number of warheads is indeed unity).

Corrective Action: At the beginning of DO loop 580, assign the actual number of warheads utilized in each target list entry to a new local parameter NWH. The number utilized is contained in parameter

(IWLB(IW,L) unless the target is an airbase; in that case the number is unity. The parameter NWH is then added to the appropriate accumulation assignment statement following labels 510, 520, or 530, respectively.

Original:

C DIVISION WEAPON SYSTEMS
C
510 CONTINUE
LIWS=IWS+NDVNW(1)*(L-1)
KWHDW(LIWS,IYL,IS)=KWHDW(LIWS,IYL,IS)+1
GO TO 580
C
SECTOR WEAPON SYSTEMS
C
520 CONTINUE
LIWS=IWS+NSCNW(1)*(L-1)
KWHSW(LIWS,IYL,IS)=KWHSW(LIWS,IYL,IS)+1
GO TO 580
C
THEATER WEAPON SYSTEMS
C
530 CONTINUE
LIWS=IWS+NTHNW(1)*(L-1)
KWHTW(LIWS,IYL,IS)=KWHTW(LIWS,IYL,IS)+1

Corrected:

DO 580 IW=1,NIW
ITC=IWLTLD(IW,L)
ISUB=IWLCOT(IW,L)
NWH=IWLB(IW,L)
IF(ITC.NE.1.AND.ISUB.EQ.1)NWH=1
KW=IWLC(IW,L)
INDEX=INDX1(KW,L)
INC=KDCDEN(INDEX,IWS,IPOS,IYL)
I JC1001
I JC1001
I JC1001
I JC1001
C
DETERMINE TYPE OF WEAPON SYSTEM
C
GO TO {510,520,530},INC
C
DIVISION WEAPON SYSTEMS
C
510 CONTINUE
LIWS=IWS+NDVNW(1)*(L-1)
KWHDW(LIWS,IYL,IS)=KWHDW(LIWS,IYL,IS)+NWH
GO TO 580 C JC1001
C
SECTOR WEAPON SYSTEMS
C
520 CONTINUE
LIWS=IWS+NSCNW(1)*(L-1)
KWHSW(LIWS,IYL,IS)=KWHSW(LIWS,IYL,IS)+NWH
GO TO 580 C JC1001
C
THEATER WEAPON SYSTEMS
C

Error C4 (Cont'd)

Corrected (Cont'd)

530 CONTINUE

LIWS=IWS+NTHNW(1)*(L-1)
KWHTW(LIWS,IYL,IS)=KWHTW(LIWS,IYL,IS)+NWH

C JC1001

C

580 CONTINUE

590 CONTINUE

Error C5: Incorrect determination of personnel casualties from latent radiation dose pools.

Location: In subroutine within DO loop 3010.

Discussion: Personnel who have not suffered immediate permanent incapacitation (IPI) due to radiation exposure will have a probability of becoming a casualty. The parameter OUT contains the fraction of personnel in a particular radiation category that have become casualties. However, in order to decrease the total number of personnel remaining within each division, this fraction is subtracted directly from the parameter PDIV instead of correctly multiplying by the factor (1-OUT).

Corrective Action: Form the factor (1-OUT) and multiply by the current value of PDIV(ID) (the actual number of personnel remaining in division labelled ID) to determine the effect of latent radiation on these remaining personnel.

Original:

```
DO 3000 IDS = 1,NTUT
  ID = IDLARA(IDS,IS)
  DO 3010 IRS = 1,4
    IF(FPRC(ID,IRS) .EQ. 0.) GO TO 3010
    OUT = FPRC(ID,IRS)*PER/TIIRD(IR)
    IF(OUT .GT. FPRC(ID,IRS))OUT = FPRC(ID,IRS)
    PDIV(ID) = PDIV(ID) - OUT
    IF(PDIV(ID) .LT. 0.) PDIV(ID) = 0.
    FPRC(ID,IRS) = FPRC(ID,IRS) - OUT
3010  CONTINUE
```

Corrected:

```
DO 3010 IRS = 1,4
  IF(FPRC(ID,IRS) .EQ. 0.) GO TO 3010
  OUT = FPRC(ID,IRS)*PER/TIIRD(IR)
  IF(OUT .GT. FPRC(ID,IRS))OUT = FPRC(ID,IRS)
  PDIV(ID)=PDIV(ID)*(1.-OUT)
  IF(PDIV(ID).LT.0.)PDIV(ID)=0.
  FPRC(ID,IRS) = FPRC(ID,IRS) - OUT
3010  CONTINUE
```

Error C6: Personnel casualties from latent radiation dose pools are not addressed for reserve divisions.

Location: In subroutine NUC6.

Discussion: The error discussed in C5 pertained only to active divisions. Reserve divisions whose personnel sustain latent radiation exposure are not even considered in casualty determination.

Corrective Action: Include the determination of casualties from latent radiation exposure for reserve divisions.

Error C6 (Cont'd)

Original:

```
3000 CONTINUE  
RETURN  
END
```

Corrected:

```
'3000 CONTINUE  
DO 3020 L=1,2  
NTOT=NDIBA(KISS,L)  
DO 3025 IDS=1,NTOT  
ID=IDLIBA(IDS,KISS,L)  
DO 3022 IRS=1,4  
IF(FPRC(ID,IRS).EQ.0.)GOTO 3022  
OUT = FPRC(ID,IRS)*PER/TTIRD(IRs)  
IF(OUT .GT. FPRC(ID,IRS))OUT = FPRC(ID,IRS)  
PDIV(ID)=PDIV(ID)*(1.-OUT)  
IF(PDIV(ID).LT.0.)PDIV(ID)=0.  
FPRC(ID,IRS) = FPRC(ID,IRS) - OUT  
3022 CONTINUE  
3025 CONTINUE  
3020 CONTINUE  
RETURN  
END
```

Error C7: Missing branch statements.

Location: In subroutine ESCLAT

- (a) following statement label 460,
- (b) following statement label 510,
- (c) following statement label 700.

Discussion: In each of the three instances, a test (comparison) is made at the designated statement label to determine whether a conditional branch statement labelled 1100 should be activated. If no conditional branch is to be made, the context of the coding requires that an unconditional branch be made to statement labelled 1200.

Corrective Action: Following each of the conditional statements, insert an unconditional branch to statement labelled 1200.

Original:

```
450 IF((L.EQ.1.AND.ISA(KS).EQ.2).AND.  
     1 (FEBA(IS)-FEBA(KS).GT.DPTH2(1))) GO TO 1100  
460 IF((L.EQ.2.AND.ISA(KS).EQ.1).AND.  
     1 (FEBA(KS)-FEBA(IS).GT.DPTH2(2))) GO TO 1100  
C   GO TO 1200  
C  
C   IS CUMULATIVE FEBA MOVEMENT PAST THRESHOLD DEPTH  
C  
500 IF(L.EQ.1.AND.FEBA(IS)-FEBATZ(IS).GT.DPTH3(1)) GO TO 1100  
510 IF(L.EQ.2.AND.FEBATZ(IS)-FEBA(IS).GT.DPTH3(2)) GO TO 1100  
C   GO TO 1200
```

Error C7 (Cont'd)

Original (Cont'd)

```

700 FRAK=QRAK{IS,L}/TQRA{IS,L}
    IF{FRAK.GT.THFRC(L)} GO TO 1100
C   GO TO 1200

```

Corrected:

```

450 IF{(L.EQ.1.AND.ISA(KS).EQ.2).AND.
     1  {FEBA{IS}-FEBA(KS).GT.DPTH2{1}}) GO TO 1100
460 IF{(L.EQ.2.AND.ISA(KS).EQ.1).AND.
     1  {FEBA(KS)-FEBA{IS}.GT.DPTH2{2}}) GO TO 1100
C   GO TO 1200                                C JC1001

C   IS CUMULATIVE FEBA MOVEMENT PAST THRESHOLD DEPTH
C
500 IF{L.EQ.1.AND.FEBA{IS}-FEBATZ{IS}.GT.DPTH3{1}} GO TO 1100
510 IF{L.EQ.2.AND.FEBATZ{IS}-FEBA{IS}.GT.DPTH3{2}} GO TO 1100 C JC1001
C   GO TO 1200

C   IS RATE OF ADVANCE SINCE LAST CYCLE BEYOND THRESHOLD DEPTH
C
600 IF{ICYCLE.EQ.1} GO TO 1200
J=0
DO 650 JE=1,NESC
IF{L.EQ.1.AND.{CFEBA{IS}.GT.DP2{JE,L}}) GO TO 620
IF{L.EQ.2.AND.{0.0-CFEBA{IS}.GT.DP2{JE,L}}) GO TO 620
GO TO 660
620 IF{J.LT.JE} J=JE
650 CONTINUE
C   DOES SIDE -L- DESIRE TO USE ESCALATION J
C
660 DO 680 ITC=1,NITC
IF{IDEL2{ITC,L}.EQ.0} GO TO 680
IF{IESC{IS,ITC,L}.LT.J} IESC{IS,ITC,L}=J
680 CONTINUE
GO TO 1200
C   IS FRACTION OF QRA A/C LOST GREATER THAN THRESHOLD
C
700 FRAK=QRAK{IS,L}/TQRA{IS,L}
    IF{FRAK.GT.THFRC(L)} GO TO 1100
C   GO TO 1200                                C JC1001

```

Error C8: Incorrect determination of sector bounds.

Location: In subroutine NDSYIN

- (a) within DO loop 700,
- (b) within DO loop 300,
- (c) within DO loop 500.

Error C8 (Cont'd)

Discussion: The three DO loops are over the range of region index IR for each combatant side' L. Within these DO loops the sector bounds (upper bound and lower bound) for the region associated within index IR are evaluated using parameter NHSR. NHSR uses a "running" index for the region in which the side 2 indices follow the side 1 indices. However, this running index was not properly generated within the DO loops.

Corrective Action: Generate the proper running index IRR corresponding to region index IR; that is

IRR = IR

IF(L.EQ.Z) IRR = IR + NR(1)

Use this running index to determine the upper and lower sector bounds for the region of interest within each DO loop.

Original:

```
DO 700 IR=1,NRS
NACTL=NAC(L)
DO 680 IAC=1,NACTL
680 TACT(IAC,IR)=0.
WTT(IR)=0.
IS1=1
IF(IR.GT.1) IS1=NHSR(IR-1)+1
IS2=NHSR(IR)
DO 650 IS=IS1,IS2
DO 300 IR=1,NRS
IF(WTT(IR).LE..0001) GO TO 300
IS1=1
IF(IR.GT.1) IS1=NHSR(IR-1)+1
IS2=NHSR(IR)
DO 355 IS=IS1,IS2
      :
      :
      :
DO 500 IR=1,NRS
IF(WTT(IR).LE..0001) GO TO 500
IS1=1
IF(IR.GT.1) IS1=NHSR(IR-1)+1
IS2=NHSR(IR)
DO 555 IS=IS1,IS2
```

Corrected:

```
DO 700 IR=1,NRS
NACTL=NAC(L)
DO 680 IAC=1,NACTL
680 TACT(IAC,IR)=0.
WTT(IR)=0.
IRR=IR
IF(L.EQ.2) IRR=IRR+NR(1)
IS1=NLSR(IRR)
IS2=NHSR(IRR)
DO 650 IS=IS1,IS2
```

I JC1001
I JC1001
C JC1001
C JC1001

Error C8 (Cont'd)

Corrected (Cont'd)

```
DO 300 IR=1,NRS
IF(WTT(IR).LE..0001) GO TO 300
IRR=IR
IF(L.EQ.2) IRR=IR+NR(1)
IS1=NLSR(IRR)
IS2=NHSR(IRR)
DO 355 IS=IS1,IS2
```

I JC1001
I JC1001
C JC1001
C JC1001

•
•
•

```
DO 500 IR=1,NRS
IF(WTT(IR).LE..0001) GO TO 500
IRR=IR
IF(L.EQ.2) IRR=IR+NR(1)
IS1=NLSR(IRR)
IS2=NHSR(IRR)
DO 555 IS=IS1,IS2
```

I JC1001
I JC1001
C JC1001
C JC1001

Error C9: Error in DO loop limits.

Location: In subroutine NRGTGS at DO loop 2660.

Discussion: The heading for DO loop 2660 has the index limits KJ = 2,NRT where NRT was previously set to the number of target supply nodes not serving the active battle area. However, the possibility of NRT = 1 was not tested, in which case the entire DO loop should be skipped (to statement labelled 2670).

Corrective Action: Before the DO statement heading, insert a conditional GOTO statement that tests whether parameter NRT is less than two and branches to statement labelled 2670 if the test is successful; that is IF(NRT.EQ.1)GOTO 2670.

Original:

```
DO 2660 KJ=2,NRT
DO 2650 JJ=1,J
IF(KRT(KJ).EQ.0) GO TO 2670
IF(KRT(KJ).EQ.ICT(JJ)) GO TO 2660
2650 CONTINUE
J=J+1
ICT(J)=KRT(KJ)
2660 CONTINUE
2670 CONTINUE
```

Corrected:

```
IF(NRT.EQ.1)GO TO 2670
DO 2660 KJ=2,NRT
DO 2650 JJ=1,J
IF(KRT(KJ).EQ.0) GO TO 2670
IF(KRT(KJ).EQ.ICT(JJ)) GO TO 2660
2650 CONTINUE
J=J+1
ICT(J)=KRT(KJ)
2660 CONTINUE
2670 CONTINUE
```

I JC1001

Error C10: Parameter ITYP not initialized.

Location: In subroutine NCZTGS.

Discussion: The subroutine PREYLD that is called within subroutine NCZTGS contains an actual parameter ITYP (the type of target subclass currently being addressed). However, no initial value is set for the parameter ITYP in subroutine NCZTGS. Since this subroutine addresses those targets within the COMMZ of the targetted side, the parameter ITYP should be set to 2, indicating rear airbases, supply nodes not feeding the active battle areas, long-range SSM sites, and rear-area divisions, respectively, for target subclasses 1, 2, 3, and 4.

Corrective Action: At the beginning of subroutine NCZTGS, initialize the parameter ITYP to two; that is ITYP = 2.

Original:

```
C DETERMINE TYPE OF COMMZ TARGET
C
C ISUB=JSTAR
C
C IF TARGET IS NOT ALLOWED AT CURRENT LEVEL OF ESCALATION GO TO NEXT
C CATEGORY OF TARGETS
C
```

Corrected:

```
C DETERMINE TYPE OF COMMZ TARGET
C
C ISUB=JSTAR
C ITYP=2
C
C IF TARGET IS NOT ALLOWED AT CURRENT LEVEL OF ESCALATION GO TO NEXT
C CATEGORY OF TARGETS
C
```

I JC1001

Error C11: Error in DO loop limits.

Location: In subroutine NCZTGS at DO loop 280.

Discussion: See Error C9.

Corrective Action: Similar to Error C9.

Original:

```
ICT(1)=KCT(1)
J=1
DO 280 KJ=2,NCT
DO 275 JJ=1,J
IF(KCT(KJ).EQ.0) GO TO 285
IF(KCT(KJ).EQ.ICT(JJ)) GO TO 280
275 CONTINUE
J=J+1
ICT(J)=KCT(KJ)
280 CONTINUE
285 CONTINUE
```

Error C11 (Cont'd)

Corrected:

```
ICT(1)=KCT(1)
J=1
IF(NCT.EQ.1)GO TO 285
DO 280 KJ=2,NCT
DO 275 JJ=1,J
IF(KCT(KJ).EQ.0) GO TO 285
IF(KCT(KJ).EQ.ICT(JJ)) GO TO 280
275 CONTINUE
J=J+1
ICT(J)=KCT(KJ)
280 CONTINUE
285 CONTINUE
```

I JC1001

Error C12: Incorrect recording of airbase values.

Location: In subroutine NUCABS in DO loop 250.

Discussion: In subroutine NUCABS within DO loop 250 the target values are scanned for airbases whose indices are held within parameter array KCT(I), the values being held in parameter array VAL(I). Upon exit from DO loop 200, the parameter IX contains the index within the array KCT corresponding to the airbase with the largest value VAL(IX), this value being held in local parameter TEMP. Following this exit from DO loop 200, a test is made to determine whether this largest airbase value is greater than zero, in which case the airbase index corresponding to this largest value is placed at the top of the current list. However, the value tested is VAL(I) instead of VAL(IX), which contains the maximum. Moreover, if the maximum value is less than or equal to zero, the local parameter KI is reset to zero and forces the formal parameter to incorrectly return a value of zero.

Corrective Action: Following DO loop 200, test the array variable VAL(IX) instead of VAL(I). Also, change the initialization of local parameter KI to contain the number of airbases having nonzero target value.

Original:

```
DO 250 I = 1,NCT
TEMP = 0.
DO 200 J = 1,NCT
IF(VAL(J) .LE. TEMP) GO TO 200
TEMP = VAL(J)
IX = J
200 CONTINUE
C
C      TEMP IS NOW THE LARGEST VALUE FROM I TO NCT. INTERCHANGE
C      THESE TWO POSITIONS THUS ASSURING THE LARGEST FIRST.
C
IF(VAL(I).GT.0.0) GO TO 260
KI=0
GO TO 280
```

Error C12 (Cont'd)

Original:

```
260 CONTINUE
ITEMP = KCT(IX)
KCT(IX) = KCT(I)
KCT(I) = ITEMP
VAL(IX) = VAL(I)
VAL(I) = TEMP
250 CONTINUE
C
C      CHECK FOR NO PEOPLE OR SHELTERS ON AIRBASE
C
```

Corrected:

```
DO 250 I = 1,NCT
IX=I
TEMP = 0.
DO 200 J = 1,NCT
IF(VAL(J) .LE. TEMP) GO TO 200
TEMP = VAL(J)
IX = J
200 CONTINUE
C
C      TEMP IS NOW THE LARGEST VALUE FROM I TO NCT. INTERCHANGE
C      THESE TWO POSITIONS THUS ASSURING THE LARGEST FIRST.
C
IF(VAL(IX).LE.0.0)GO TO 260
C   JC1001
C      KI=0
C   JC1001
C      GO TO 280
C   JC1001
ITEMP = KCT(IX)
KCT(IX) = KCT(I)
KCT(I) = ITEMP
VAL(IX) = VAL(I)
VAL(I) = TEMP
250 CONTINUE
C
C      CHECK FOR NO PEOPLE OR SHELTERS ON AIRBASE
C
```

Error C13: Incorrect number of weapons assigned (NWPN).

Location: In subroutine PREYLD after statement labelled 400.

Discussion: The local parameter NWPN that contains the number of weapons to be assigned to the current target list is initialized to one following the statement labelled 400 for SSM-type targets. However, the formal parameter KRT contains the number of SSM sites targetted and may contain a number greater than one.

Corrective Action: Following the statement labelled 400, set the parameter NWPN equal to KRT to correctly specify the number of weapons needed for the SSM targets.

Error C13 (Cont'd)

Original:

```
C      SURFACE TO SURFACE MISSILE SITES
C
C 400 CONTINUE
  YPRE=YLTS(M(L)
  NWPN=1
C  DIST=*****
  DIST=75
  GO TO 1000
```

Corrected:

```
NWPN=KRT
C  DIST=*****
  DIST=75
C
  GO TO 1000
```

Error C14: Improper decrease in nuclear warhead inventory.

Location: In subroutine PREYLD following statement labelled 2420.

Discussion: Following statement labelled 2420 after a weapons system has been designated for a particular target, the parameter NIWAS is incremented and then tested to determine whether it exceeds 100. If so, the target list is considered saturated and the current target is not included in the list. However, the weapon delivery system and warhead inventories have previously been adjusted to account for the weapons system selection.

Corrective Action: Perform the test to determine whether NIWAS exceeds 100 at the beginning of subroutine PREYLD before weapons and warhead inventory adjustments.

Original:

```
2420 CONTINUE
  NIWAS=NIWAS+1
C
C TEST TO SEE IF AVAILABLE ARRAY SIZE IS EXCEEDED
C
C
  IF(NIWAS.GT.100) GO TO 5000
```

Corrected:

```
DIMENSION IPAAC(6)
C
C TEST TO SEE IF AVAILABLE ARRAY SIZE IS EXCEEDED
C
C
  IF(NIWAS.GE.100) GO TO 5000
C
  NWPN=1
```

M JCI001

Error C15: Incorrect combatant side index.

Location: In subroutine PREYLD following statement labelled 2420

- (a) adjustment to parameter NWABA,
- (b) adjustment to parameter NWREG,
- (c) adjustment to parameter NWCZ.

Discussion: The parameters, NWABA, NWREG, and NWCZ, contain the numbers of warheads (weapons) received by the targetted side in the active battle area, the rear area, and the COMM2, respectively. These parameters are later used to update the escalation state of the targetted side. However, the combatant side index in the statements used to adjust these parameters in subroutine PREYLD is L, which is not the targetted side index but is, instead, the targetting side index.

Corrective Action: At the beginning of subroutine PREYLD, specify the targetted side index K; that is,

K - 3 - L,

and change the index L to K in the statements following labelled statement 2420 that adjust the three parameters, NWABA, NWREG, and NWCZ.

Original:

```
C      INCREMENT COUNT OF TARGETS ATTACKED
C      NWABA{IS,L} = NWABA{IS,L} + NWPN
C
C      GO TO 5000
C
2410 CONTINUE
      IWLDST{I,L}=KRT
      IF{ISUB.EQ.4} IWLDST{I,L}=IDV
      IWLCUT{I,L}=ISUB
      IWLTZN{I,L}=ITYP
      IF{ISUB.EQ.4} IWLTZN{I,L}=ISU
C
C      INCREMENT COUNT OF TARGETS ATTACKED
C
C      * IF{ITC .EQ. 3} GO TO 2412
C
C      AND --ONE-- WEAPON TO USAGE
C
      NWREG{ISUB,IS,L} = NWREG{ISUB,IS,L} + 1
      GO TO 2413
2412 CONTINUE
      NWCZ{ISUB,IS,L} = NWCZ{ISUB,IS,L} + 1
2413 CONTINUE
```

Error C15 (Cont'd)

Corrected:

NWPN=1
IS=KISS
K=3-L

I JC1001

•
•
•
C INCREMENT COUNT OF TARGETS ATTACKED
NWABA(IS,K) = NWABA(IS,K) + NWPN C JC1001
C GO TO 5000
C
2410 CONTINUE
IWLDIS(I,L)=KRT
IF(IISUB.EQ.4) IWLDIS(I,L)=IDV
IWLCOT(I,L)=ISUB
IWLTZN(I,L)=ITYP
IF(IISUB.EQ.4) IWLTZN(I,L)=ISU
C INCREMENT COUNT OF TARGETS ATTACKED
C IF(ITE .EQ. 3) GO TO 2412
C ADD --ONE-- WEAPON TO USAGE
C
NWREG(IISUB,IS,L) = NWREG(IISUB,IS,L) + NWPN C JC1001
GO TO 2413
2412 CONTINUE
NWCZ(IISUB,IS,L) = NWCZ(IISUB,IS,L) + NWPN C JC1001
2413 CONTINUE

Error C16: Incorrect adjustment to parameters NWREG and NWCZ.

Location: In subroutine PREYLD following statements labelled 2410 and 2412.

Discussion: The array variables NWREG and NWCZ contain the number of weapons (warheads) received by the targetted side in the rear area and COMMZ, respectively. These variables are incremented by unity for each target type independent of the number of targets actually addressed within each type. Thus, more than one SSM site may be targetted and more than one division subunit of a specified type may be targetted. The increment in NWREG and NWCZ should reflect this by using the variable NWPN that contains the number of warheads and weapons assigned to each target type.

Corrective Action: In the statements adjusting the parameters NWREG and NWCZ, increment these parameters by the variable NWPN.

Original:

C ADD --ONE-- WEAPON TO USAGE
C
NWREG(IISUB,IS,L) = NWREG(IISUB,IS,L) + 1
GO TO 2413
2412 CONTINUE
NWCZ(IISUB,IS,L) = NWCZ(IISUB,IS,L) + 1
2413 CONTINUE

Error C16 (Cont'd)

Corrected:

```
C ADD --ONE-- WEAPON TO USAGE
C
C NWREG(ISUB,IS,K) = NWREG(ISUB,IS,K) + NWPN          C JC1001
GO TO 2413
2412 CONTINUE
NWCZ(ISUB,IS,K) = NWCZ(ISUB,IS,K) + NWPN          C JC1001
2413 CONTINUE
```

SECTION D - GROUND, AIR-GROUND, AND THEATER CONTROL SIMULATION

Error D1: Incorrect parameter used for air munitions.

Location: In subroutine GC before DO loop 430.

Discussion: The local parameter N5 is set to NW(L) instead of NAM(L). N5 is later used as an index limit in DO loops 410 and 425 to operate over the number of air munitions available to side L.

Corrective Action: Change the parameter NW(L) to NAM(L) in the assignment statement for parameter N5.

Original:

```
C -----
C 4) ADJUST THE STANDARD ALLOCATION.
C -----
C THIS SECTION EXECUTED ONLY ON FIRST CYCLE OF GAME
IF(ICYCLE.NE.1) GO TO 499
DO 450 L=1,2
N3=NW(L)
K=3-L
N4=NW(K)
N5=NW(L)
DO 430 KW=1,N4
IF(PWSF(KW,K).EQ.0.0) GO TO 415
DO 405 IW=1,N3
SAWA(IW,KW,L)=SAWA(IW,KW,L)/PWSF(KW,K)
SAWD(IW,KW,L)=SAWD(IW,KW,L)/PWSF(KW,K)
405 CONTINUE
```

Corrected:

```
C -----
C 4) ADJUST THE STANDARD ALLOCATION.
C -----
C THIS SECTION EXECUTED ONLY ON FIRST CYCLE OF GAME
IF(ICYCLE.NE.1) GO TO 499
DO 450 L=1,2
N3=NW(L)
K=3-L
N4=NW(K)
NS=NAM(L)
DO 430 KW=1,N4          C JC1001
```

Error D1 (Cont'd)

Corrected:

```
IF(PWSF(KW,K).EQ.0.0) GO TO 415  
DO 405 IW=1,N3  
SAWA(IW,KW,L)=SAWA(IW,KW,L)/PWSF(KW,K)  
SAWD(IW,KW,L)=SAWD(IW,KW,L)/PWSF(KW,K)  
405 CONTINUE
```

Error D2: Incorrect position of statement for initialization of parameter FMS.

Location: In subroutine FEBAMT before statement label 1004.

Discussion: Statement label 1004 allows a loop back from the conditional branch following statement label 1569 in the event that the forward edge of battle area (FEBA) movement crosses an interval boundary. Within the loop-back section, the parameter FMS associated with the average mobility factor for side L in sector IS is recalculated for the new interval parameters. However, FMS is not reinitialized to zero before this recalculation.

Corrective Action: Place the initialization statement for FMS at or after the statement label 1004.

Original:

```
PTINI=1.  
CFEBA1=0.  
C FMS = AVE. MOBILITY FACTOR FOR SECTOR IS  
FMS=0.  
ISTOP=0  
1004 KP=KPS(IIS)  
IF(KP.EQ.4) KP=2  
IF(KP.EQ.5) KP=1  
KT=KTER(IIS)  
N1=1  
N2=NDS(IIS,1)  
IF(L.EQ.1) GO TO 1005  
N1=1+N2  
N2=N2+NDS(IIS,2)  
1005 DO 1010 IDS=N1,N2
```

Corrected:

```
1004 KP=KFS(IIS)  
IF(KF.EQ.4) KP=2  
IF(KP.EQ.5) KP=1  
KT=KTEF(IIS)  
C FMS = AVE. MOBILITY FACTOR FOR SECTOR IS M JC1001  
FMS=0. M JC1001  
IF(NDS(IIS,L).EQ.0) GOTO 1011 I JC1001  
N1=1  
N2=NDS(IIS,1)  
IF(L.EQ.1) GO TO 1005  
N1=1+N2  
N2=N2+NDS(IIS,2)  
1005 DO 1010 IDS=N1,N2
```

Error D3: Possible divide-by-zero situation.

Location: In subroutine FEBAMT following DO loop 1010.

Discussion: The parameter NDS(IS,L) is used to set and adjust the index limits for DO loop 1010 and is also used immediately following the DO loop as a divisor in the assignment statement for local parameter FMS. The possibility exists for NDS(IS,L) to have the value zero, in which case the DO loop 1010 and the assignment statement for FMS should be bypassed.

Corrective Action: Perform a conditional branch to the statement following the assignment statement for FMS, if the parameter NDS(IS,L) = 0.

Original:

```
ID=IDLABA(IDS,IS)
IT=ITD(ID)
FMS=FMS+FMDPT(IT,KP,KT)
1010 CONTINUE
FMS=FMS/NDS(IS,L)
(
IF(IPRD.NE.1) GO TO 1500
WRITE(MOT,110)
WRITE(MOT,22) FMS
```

Corrected:

```
IF(NDS(IS,L).EQ.0)GOTO 1011 I JCI001
N1=1
N2=NDS(IS,1)
IF(L.EQ.1) GO TO 1005
N1=1+N2
N2=N2+NDS(IS,2)
1005 DO 1010 IDS=N1,N2
ID=IDLABA(IDS,IS)
IT=ITD(ID)
FMS=FMS+FMDPT(IT,KP,KT)
1010 CONTINUE
FMS=FMS/NDS(IS,L)
1011 CONTINUE I JCI001
```

Error D4: Incorrect logical operator used.

Location: In subroutine FEBAMT within DO loop 1534.

Discussion: DO loop 1534 is used to accumulate the weapons systems values VALD for the defending side divisions located in the first inactive battle areas above and below the sector IS currently being processed. However, the conditional branch statement within DO loop 1534 incorrectly uses the logical operator .OR. instead of the operator .AND. to exclude divisions not properly located.

Corrective Action: Change the logical operator .OR. to the operator .AND..

Error D4 (Cont'd)

Original:

```
C BOTH FLANKS HAVE EXCESS EXPOSURE, COMPUTE DEFENDER TOTAL GROUND
C VALUE IN FIRST INACTIVE BATTLE AREAS OF BOTH ADJACENT SECTORS
C (VALD)
1530 JBA=IABAS{IS-1}+ISIGN*NS
LBA=IABAS{IS+1}+ISIGN*NS
VALD=0.
DO 1534 ID=N1D,N2D
IF(JBA.NE.IBALD{ID}.OR.LBA.NE.IBALD{ID}) GO TO 1534
VALD=VALD+VDDSF{ID}
1534 CONTINUE
GO TO 1550
```

Corrected:

```
C BOTH FLANKS HAVE EXCESS EXPOSURE, COMPUTE DEFENDER TOTAL GROUND
C VALUE IN FIRST INACTIVE BATTLE AREAS OF BOTH ADJACENT SECTORS
C (VALD)
1530 JBA=IABAS{IS-1}+ISIGN*NS
LBA=IABAS{IS+1}+ISIGN*NS
VALD=0.
DO 1534 ID=N1D,N2D
IF(JBA.NE.IBALD{ID}.AND.LBA.NE.IBALD{ID}) GO TO 1534 C JC1001
VALD=VALD+VDDSF{ID}
1534 CONTINUE
GO TO 1550
```

Error D5: Inconsistent sign convention used for CFEBA determination.

Location: In subroutine FEBAMT after statement labelled 1565.

Discussion: In the parts of subroutine FEBAMT that precede line labelled 1565, the parameter CFEBA(IS) associated with the change in FEBA in sector IS has a positive value no matter which combatant side is the attacker. Similarly, the parameter CFEBAL that contains the cumulative FEBA change across one or more intervals is also positive independent of attacking side. However, the final evaluation of the parameter CFEBA(IS) in the statement following 1565 inconsistently contains a term whose sign depends on the attacker.

Corrective Action: In the final evaluation of CFEBA(IS) multiply both terms CFEBAL and CFEBA(IS)*PTINI by the common sign parameter ASIGN.

Original:

```
CFEBA{IS}=CFEA1+CFEA{IS}*PTINI*ASIGN
1570 CONTINUE
C CALCULATIONS COMPLETED FOR THEATER ATTACKER, SET SIDE L = THEATER
C DEFENDER AND COMPUTE FEBA C. A/I/C FOR DEFENDER ON ATTACK
L=3-ITA
1575 CONTINUE
```

Error D5 (Cont'd)

Corrected:

```
C      FEBA MOVEMENT CALCULATIONS HAVE BEEN COMPLETED FOR IS, SET FEBA AT
C      COMPUTED LOCATION(CTEMP), COMPUTE TOTAL FEBA MOVEMENT(CFEBA1)
1565  FEBA1(IS)=CTEMP
      CFEBA1(IS)=(CFEEA1+CFEBA1(IS)*PTINI)*ASIGN          C JC1001
1570  CONTINUE
C      CALCULATIONS COMPLETED FOR THEATER ATTACKER, SET SIDE L = THEATER
C      DEFENDER AND COMPUTE FEBA CHANGE FOR DEFENDER ON ATTACK
L=3-ITA
1575  CONTINUE
```

Error D6: Code is bypassed by omission of detailed printout.

Location: In subroutine FEBAMT following DO loop 1575.

Discussion: The conditional branch statement for omitting detailed printout following DO loop 1575 causes a branch to statement labelled 1800 and effectively bypasses the FEBA movement calculations for holding posture when the defending side withdraws.

Corrective Action: Change the conditional branch statement to branch to a new label (that is, 1599) at the beginning of the previously bypassed section number 17.

Original:

```
C
      IF(IPRD.NE.1) GO TO 1800
      #WRITE(MOT,111)
      #WRITE(MOT,22) (CFEBA1(IS),IS=1,NS)
      #WRITE(MOT,112)
      #WRITE(MOT,22) (FEBA1(IS),IS=1,NS)
      #WRITE(MOT,118)
      #WRITE(MOT,10) (INTRVL(IS),IS=1,NS)
      #WRITE(MOT,119)
      #WRITE(MOT,10) (KPS1(IS),IS=1,NS)
      #WRITE(MOT,120)
      #WRITE(MOT,10) (KTER1(IS),IS=1,NS)
C -----
C 17) WITHDRAW DUE TO COMBAT DEPLOYMENT.
C -----
C      WITHDRAW FEBA FOR DEFENDER DEPLOYMENT CHANGES IN SECTORS WHERE
C      THERE IS A HOLDING SITUATION
      ASIGN=-1
      DO 1720 L=1,2
      DO 1710 IS=1,NS
      IF(I#DRAN1(IS,L).EQ.0) GO TO 1710
      IF(I$AI1(IS).GT.0) GO TO 1710
      CFEBA1(IS)=ASIGN*DDWDCD(L)
      FEBA1(IS)=FEBAL1(IS)-CFEBA1(IS)
1710  CONTINUE
      ASIGN=1
1720  CONTINUE
C
```

Error D6 (Cont'd)

Original:

```
IF(IPRD.NE.1) GO TO 1799
WRITE(MOT,111)
WRITE(MOT,22) (CFEBA(IS),IS=1,NS)
WRITE(MOT,112)
WRITE(MOT,22) (FEBA(IS),IS=1,NS)
1799 CONTINUE
C -----
C 18) DETERMINE IF THEATER ATTACKER CHANGES FOR NEXT CYCLE.
C -----
C THEATER DEFENDER BECOMES THEATER ATTACKER IF CURRENT TA NOT
C ATTACKING OR HOLDING DUE TO CONSTRAINED FLANK IN AT LEAST 1
C SECTOR
C IFLAG = INDEX TO THEATER ATTACKER NEXT CYCLE
1800 IFLAG=ITA
```

Corrected:

```
IF(IPRD.NE.1) GO TO 1599
WRITE(MOT,111) C JC1001
WRITE(MOT,22) (CFEBA(IS),IS=1,NS)
WRITE(MOT,112)
WRITE(MOT,22) (FEBA(IS),IS=1,NS)
WRITE(MOT,118)
WRITE(MOT,10) (INTRVL(IS),IS=1,NS)
WRITE(MOT,119)
WRITE(MOT,10) (KPS(IS),IS=1,NS)
WRITE(MOT,120)
WRITE(MOT,10) (KTER(IS),IS=1,NS)
1599 CONTINUE I JC1001
C -----
C 17) WITHDRAW DUE TO COMBAT DEPLOYMENT.
C -----
C WITHDRAW FEBA FOR DEFENDER DEPLOYMENT CHANGES IN SECTORS WHERE
C THERE IS A HOLDING SITUATION
ASIGN=1 C JC1001
DO 1720 L=1,2
DO 1710 IS=1,NS
IF(IWDRAW(IS,L).EQ.0) GO TO 1710
IF(ISA(IS).GT.0) GO TO 1710
CFEBA(IS)=ASIGN*DWDCD(L)
FEBA(IS)=FFEA(IS)+CFEBA(IS) C JC1001
1710 CONTINUE
ASIGN=-1 C JC1001
1720 CONTINUE
```

Error D7: Inconsistent use of sign for parameter CFEBA.

Location: In subroutine FEBAMT within DO loop 1710.

Discussion: Within DO loop 1710 the parameter CFEBA(IS) is determined for those sectors IS which are in a holding posture and where the theater defender has elected to withdraw. However, the sign given to the parameter CFEBA(IS) is positive for side 2 withdrawing

Error D7 (Cont'd)

Original (Cont'd)

and negative for side 1 withdrawing. These sign conventions are inconsistent with the conventions in preceding sections that have CFEBA(IS) positive for side 2 advancing and negative for side 1 advancing.

Corrective Action: (a) Initially set the parameter ASIGN = +1 before DO loop 1720 and then to ASIGN = -1 following DO loop 1710.

(b) Change the sign in the assignment statement for the parameter FEBA(IS) within DO loop 1710 [th: i.e., FEBA(IS) = FEBA(IS) + CFEBA(IS)].

Original:

```
C -----
C 17) WITHDRAW DUE TO COMBAT DEPLOYMENT.
C -----
C      WITHDRAW FEBA FOR DEFENDER DEPLOYMENT CHANGES IN SECTORS WHERE
C      THERE IS A HOLDING SITUATION
ASIGN=1                                     C JC1001
DO 1720 L=1,2
DO 1710 IS=1,NS
IF(IWDRAW(IS,L).EQ.0) GO TO 1710
IF(ISA(IS).GT.0) GO TO 1710
CFEBA(IS)=ASIGN,=DDWDCD(L)
FEBA(IS)=FEBA(IS)+CFEBA(IS)                 C JC1001
1710 CONTINUE
ASIGN=-1                                     C JC1001
1720 CONTINUE
```

Corrected:

```
C -----
C 17) WITHDRAW DUE TO COMBAT DEPLOYMENT.
C -----
C      WITHDRAW FEBA FOR DEFENDER DEPLOYMENT CHANGES IN SECTORS WHERE
C      THERE IS A HOLDING SITUATION
ASIGN=1                                     C JC1001
DO 1720 L=1,2
DO 1710 IS=1,NS
IF(IWDFAW(IS,L).EQ.0) GO TO 1710
IF(ISA(IS).GT.0) GO TO 1710
(FEBA(IS)=ASIGN,=DDWDCD(L)
FEBA(IS)=FEBA(IS)+FEBA(IS))                  C JC1001
1710 CONTINUE
ASIGN=-1                                     C JC1001
1720 CONTINUE
```

Error D8: Local utility parameter TEMP1 not initialized.

Location: In subroutine FEBAMT following statement labelled 2004.

Discussion: The local parameter TEMP1 that later holds the value of flank exposure on a sector's lower boundary is not initialized to zero.

Error D8 (Cont'd)

Corrective Action: Initialize the parameter TEMP1 to zero following the statement labelled 2004 (that is, TEMP1 = 0).

Original:

```
C IF THERE IS MORE THAN 1 SECTOR, COMPUTE AMOUNT OF EXPOSURE OF EACH
C FLANK(TEMP,TEMP1) (FIRST AND LAST SECTORS HAVE ONLY 1 FLANK)
2004 IF{NS.EQ.1} GO TO 2020
    TEMP=0.
    IF{IS.EQ.1} GO TO 2005
    TEMP=FEBA{IS-1}-FEBA{IS}
    IF{IS.EQ.NS} GO TO 2006
2005 TEMP1=FEBA{IS+1}-FEBA{IS}
2006 IF{L.NE.2} GO TO 2010
    TEMP=-TEMP
    TEMP1=-TEMP1
C GO ON TO NEXT SECTOR IF EXPOSURE OF BOTH FLANKS ACCEPTABLE
```

Corrected:

```
C IF THERE IS MORE THAN 1 SECTOR, COMPUTE AMOUNT OF EXPOSURE OF EACH
C FLANK(TEMP,TEMP1) (FIRST AND LAST SECTORS HAVE ONLY 1 FLANK)
2004 IF{NS.EQ.1} GO TO 2020
    TEMP=0.
    TEMP1=0.
    IF{IS.EQ.1} GO TO 2005
    TEMP=FEBA{IS-1}-FEBA{IS}
    IF{IS.EQ.NS} GO TO 2006
2005 TEMP1=FEBA{IS+1}-FEBA{IS}
2006 IF{L.NE.2} GO TO 2010
    TEMP=-TEMP
```

? JC1001

Error D9: Incorrect combatant side index employed.

Location: In subroutine FEBAMT within statement labelled 2106.

Discussion: Statement labelled 2106 is a conditional branch to statement labelled 2150 when a sector has no excessive flank exposure for the theater attacker combatant side (given by side index ITA). However, within statement 2106 the side index for parameter FEAFBA is incorrectly set to L.

Corrective Action: Change the side index from L to ITA within parameter FEAFBA.

Original:

```
TEMP1=0.
TEMP2=0.
IF{IS.EQ.1} GO TO 2105
TEMP1=ASIGN*FEBA{IS}-ASIGN*FEBA{IS-1}
IF{IS.EQ.NS} GO TO 2106
2105 TEMP2=ASIGN*FEBA{IS}-ASIGN*FEBA{IS+1}
C GO ON TO NEXT SECTOR IF EXPOSURE OF BOTH FLANKS ACCEPTABLE
```

Error D9 (Cont'd)

Original (Cont'd)

```
2106 IF(TEMP1.LE.FEAFBA(ITA).AND.TEMP2.LE.FEAFBA(L)) GO TO 2150
      IF(IUSFRC.EQ.1) GO TO 2110
C     SECURITY FORCE RATIO NOT USED, TA IS CONSTRAINED IN IS
2108 ISCEF(IS)=1
      GO TO 2150
```

Corrected:

```
TEMP1=0.
TEMP2=0.
IF(IS.EQ.1) GO TO 2105
TEMP1=ASIGN*FEBA(IS)-ASIGN*FEBA(IS-1)
IF(IS.EQ.NS) GO TO 2106
2105 TEMP2=ASIGN*FEBA(IS)-ASIGN*FEBA(IS+1)
C     GO ON TO NEXT SECTJR IF EXPOSURE OF BOTH FLANKS ACCEPTABLE
2106 IF(TEMP1.LE.FEAFBA(ITA).AND.TEMP2.LE.FEAFBA(ITA)) GO TO 2150 C JCI001
      IF(IUSFRC.EQ.1) GO TO 2110
C     SECURITY FORCE RATIO NOT USED, TA IS CONSTRAINED IN IS
2108 ISCEF(IS)=1
      GO TO 2150
```

Error D10: Incorrect logical operator used.

Location: In subroutine FEBAMT within DO loop 2134.

Discussion: Same as Error D4.

Corrective Action: Similar to Error D4.

Original:

```
C     COMPUTE DEFENDER GROUND VALUE IN CONSTRAINING SECTOR(S) (VALD)
      IF(TEMP1.GT.FEAFBA(ITA).AND.TEMP2.GT.FEAFBA(ITA)) GO TO 2130
      IF(TEMP1.GT.FEAFBA(ITA)) GO TO 2135
      GO TO 2136
2130 JBA=IABAS(IS-1)+ASIGN*NS
      LBA=IABAS(IS+1)+ASIGN*NS
      VALD=0.
      DO 2134 ID=N1D,N2D
      IF(JBA.NE.IBALD(ID).OR.LBA.NE.IBALD(ID)) GO TO 2134
      VALD=VALD+VCDSF(ID)
2134 CONTINUE
      GO TO 2145
```

Corrected:

```
C     COMPUTE DEFENDER GROUND VALUE IN CONSTRAINING SECTOR(S) (VALD)
      IF(TEMP1.GT.FEAFBA(ITA).AND.TEMP2.GT.FEAFBA(ITA)) GO TO 2130
      IF(TEMP1.GT.FEAFBA(ITA)) GO TO 2135
      GO TO 2136
2130 JBA=IABAS(IS-1)+ASIGN*NS
      LBA=IABAS(IS+1)+ASIGN*NS
      VALD=0.
      DO 2134 ID=N1D,N2D
```

Error D10 (Cont'd)

Corrected (Cont'd)

```
IF(JBA.NE.IBALD(ID).AND.LBA.NE.IBALD(ID)) GO TO 2134      C JC1001
VALD=VALD+VDDSF(ID)
2134 CONTINUE
GO TO 2145
```

Error D11: Incorrect reallocation of excess SSM attack aircraft.

Location: In subroutine ASGATR

- (a) before statement labelled 189,
- (b) within DO loop 188.

Discussion: (a) In this section the local parameter TNTA contains the number of SSM targets that can be attacked by interdiction aircraft. Similarly, the local parameter USE contains the fraction of SSM targets that can be interdicted, and the parameter Q contains the fraction of SSM targets that cannot be interdicted (that is, $Q = 1 - USE$). However, the parameter Q is later used as if it contained the fraction of aircraft that are not required for SSM interdiction.

(b) Within DO loop 188 the conditional branch statement IF(USE.GE.1.)GO TO 19 is invalid.

Corrective Action: (a) The assignment statements for the parameter Q should be changed to be consistent with later usage, that is,

$Q = 0$

IF(TNTA.GE.T)Q = 1. - TNTA * RSIZE/S11 .

Thus, Q is zero if the total number of SSM targets that can be attacked is less than T--the total number of SSM targets located (i.e., no excess interdiction raids are available). However, if excess aircraft exist (i.e., TNTA = T), then Q is finite.

(b) Eliminate the conditional branch statement.

Original:

```
USE=TNTA/T
Q=1.-USE
Z(1)=(PDSSMS(L)*SSMSFS(1,IS,KL)/T)*USE
Z(2)=(PDSSMS(L)*SSMSFS(2,IS,KL)/T)*USE
189 CONTINUE
S12=0.
S13=0.
DO 188 IT=1,KAC
IF(TNTA.LE.0.)GOTO190
C     SSM ATTRITION
IF(USE.GE.1.)GOTO19
C     HERE, THERE EXISTS AN XS OF A/C OVER SSM SITES, SO APPORTION
C     TO OTHER MISSIONS
```

Error D11 (Cont'd)

Corrected:

```
Q=0.  
IF(TNTA.GE.T)Q=1.-TNTA+RSIZE/S11  
Z(1)=(PDSSMS(L)+SSMSFS(1,IS,KL)/T)*USE  
Z(2)=(PDSSMS(L)+SSMSFS(2,IS,KL)/T)*USE  
189 CONTINUE  
S12=0.  
S13=0.  
DO 158 IT=1,KAC  
IF(TNTA.LE.0.)GOTO190  
C JCI IF(USE.GE.1.)GOTU19  
C SSM ATTRITION  
C HERE, THERE EXISTS AN XS OF A/C OVER SSM SITES, SO APPORTION  
C TO OTHER MISSIONS
```

I JC1001
C JC1001

I JC1001
C JC1001

Error D12: Incorrect reallocation of aircraft to rear area divisions.

Location: In subroutine ASGATR within DO loop 188.

Discussion: In this section the reallocation of excess SSM interdiction raids to both rear area divisions and supply nodes occurs regardless of the possibility that no rear area divisions exist. To be consistent with previous reallocations of interdiction aircraft, a check should be made to determine whether any rear area divisions exist (given by the local parameter NOD). If none exist, then all excess aircraft should be reallocated to interdict supply nodes.

Corrective Action: With DO loop 188, perform a conditional branch (upon condition NOD = 0) to reallocate all excess interdiction aircraft supply node targets.

Original:

```
C HERE, THERE EXISTS AN XS.OF A/C OVER SSM SITES, SO APPORTION  
C TO OTHER MISSIONS  
AS=Q*V(1,IT)  
V(1,IT)=USE*V(1,IT)  
S1=PIAIM(IT,2,L)+PIAIM(IT,3,L)  
IF(S1.LE.0.)S1=1.  
V(2,IT)=V(2,IT)+(PIAIM(IT,2,L)/S1)*XS  
V(3,IT)=V(3,IT)+(PIAIM(IT,3,L)/S1)*XS  
19 GOTO 20 I=1,2  
Q2=Z(1)*1  
IF(Q2.GT.PKASSM(IT,I,L)) GO TO 201  
P(I)=0.  
GO TO 20  
201 IF(Z(1).GT.0.) P(I)=P(I)+(1.-PKASSM(IT,I,L)/Q2)*Z(1)*V(1,IT)  
20 CONTINUE
```

Error D12 (Cont'd)

Corrected:

```

C HERE, THERE EXISTS AN XS OF A/C OVER SSM SITES, SO APPORTION
C TO OTHER MISSIONS
XS=Q*V(1,IT)
V(1,IT)=V(1,IT)-XS
IF(NOD.LE.0)GOTO 17
S1=PIAIM(IT,2,L)+PIAIM(IT,3,L)
IF(S1.LE.0.)S1=1.
V(2,IT)=V(2,IT)+(PIAIM(IT,2,L)/S1)*XS
V(3,IT)=V(3,IT)+(PIAIM(IT,3,L)/S1)*XS
GOTO 19
17 V(3,IT)=V(3,IT)+XS
19 DO 20 I=1,2
QZ=Z(I)*T
IF(QZ.GT.PYASSM(IT,I,L)) GO TO 201
P(I)=U.
GO TO 20
201 IF(Z(I).GT.0.) P(I)=P(I)+(1.-PKASSM(IT,I,L)/QZ)*(Z(I)*V(I,IT))
20 CONTINUE

```

C JC1001
I JC1001

I JC1001
I JC1001

Error D13: Incorrect assignment to region index.

Location: In subroutine AIRGRD before DO loop 60020.

Discussion: Within DO loop 60020 the region index IR is used in two parameters FAAAFF and FAAARF. However, the parameter IR has been incorrectly set to a "running" index value

[that is, IR = IRT = IR = (L-1) * NR(1)].

Corrective Action: Eliminate the assignment statement for parameter IR preceding DO loop 60020.

Original:

```

60019 CONTINUE
IR=IRT
DO 60020 KS=JLS,JHS
I=KR+NR(1)*(L-1)
SUM=SUM+FAAAFF(IAC,IR,I)*ACFS(IAC,KS,L)
SUM1=SUM1+FAAARF(IAC,IR,I)*ACRSE(IAC,KS,L)
60020 CONTINUE

```

Corrected:

```

60019 CONTINUE
C JC1 IR=IRT
DO 60020 KS=JLS,JHS
I=KR+NR(1)*(L-1)
SUM=SUM+FAAAFF(IAC,IR,I)*ACFS(IAC,KS,L)
SUM1=SUM1+FAAARF(IAC,IR,I)*ACRSE(IAC,KS,L)
60020 CONTINUE

```

C JC1001

Error D14: Incorrect assignment to region index.

Location: In subroutine AIRGRD before DO loop 62020.

Discussion: Similar to Error D13.

Corrective Action: Similar to Error D13.

Original:

```
62019 CONTINUE
  IR=IRT
  DO 62020 KS=JLS,JHS
    I=KR+NR(1)+(L-1)
    SUM=SUM+FAAAFF(IAC,IR,I)*ACFS(IAC,KS,L)
    SUM1=SUM1+FAAARF(IAC,IR,I)*ACRS(IAC,KS,L)
62020 CONTINUE
```

Corrected:

```
62019 CONTINUE
  IR=IRT
  DO 62020 KS=JLS,JHS
    I=KR+NR(1)+(L-1)
    SUM=SUM+FAAAFR(IAC,IR,I)*ACFS(IAC,KS,L) C JCI001
    SUM1=SUM1+FAAARR(IAC,IR,I)*ACRS(IAC,KS,L) C JCI001
62020 CONTINUE C JCI001
```

Error D15: Incorrect allocation parameters used.

Location: In subroutine AIRGRD

- (a) within DO loop 62020
- (b) following DO loop 62020.

Discussion: The section containing DO loop 62020 (and following) determines the number of aircraft allocated to attack enemy rear-region airbases. However, the allocation parameters FAAAFF, FAAARF, and FAAAZF are associated with allocations to enemy forward-region airbases.

Corrective Action: Exchange the parameters FAAAFR, FAAARR, and FAAAZR for the parameters FAAAFF, FAAARF, and FAAAZF, respectively.

Original:

```
DO 62020 KS=JLS,JHS
  I=KR+NR(1)+(L-1)
  SUM=SUM+FAAAFF(IAC,IR,I)*ACFS(IAC,KS,L)
  SUM1=SUM1+FAAARF(IAC,IR,I)*ACRS(IAC,KS,L)
62020 CONTINUE
  SUM2=FAAAZF(IAC,KR,L)*ACCZ(IAC,L)
```

Error D15 (Cont'd)

Corrected:

```
DO 62020 KS=JLS,JHS
I=KR+NR(1)*(L-1)
SUM=SUM+FAAAFR(IAC,IR,I)*ACFS(IAC,KS,L)
SUM1=SUM1+FAAARR(IAC,IR,I)*ACRS(IAC,KS,L)          C JCI001
62020 CONTINUE                                         C JCI001
SUM2=FAAAZR(IAC,KR,L)*ACCZ(IAC,L)                  C JCI001
```

Error D16: Incorrect accumulation of aircraft attacking COMMZ.

Location: In subroutine AIRGRD within DO loop 64015.

Discussion: Within DO loop 64015 the local parameters SUM, SUM1, and SUM2 are used to accumulate the total number of aircraft of type IAC that are allocated to attack enemy COMMZ airbases. Specifically, DO loop 64005 contains a conditional branch statement using an undefined sector parameter IS which is irrelevant for COMMZ airbases.

Corrective Action: Within and following DO loop 64005:

- (a) Eliminate the conditional branch to statement labelled 64006.
- (b) Eliminate the statement labelled 64006.
- (c) Nest the DO loop 64010 inside DO loop 64005.
- (d) Eliminate the superfluous assignment statement for index IR.

Original:

```
C      SUM AIRCRAFT OVER ALL SECTOR CONTAINED IN SAME REGION AS SECTOR
C      IS FOR ATTACKER
C
JR=NR(L)
DO 64005 IR=1, JR
IRT=IR+(L-1)*NR(1)
JLS=NLSR(IRT)
JHS=NHSR(IRT)
IF(IS.GE.JLS.AND.IS.LE.JHS) GO TO 64006
64005 CONTINUE
64006 CONTINUE
C
IR=IRT
DO 64010 KS=.ILS,JHS
SUM=SUM+FAAAFZ(IAC,IR,L)*ACFS(IAC,KS,L)
SUM1=SUM1+FAAARZ(IAC,IR,L)*ACRS(IAC,KS,L)
64010 CONTINUE
SUM2=FAAAZZ(IAC,L)*ACCZ(IAC,L)
```

Error D16 (Cont'd)

Corrected:

```
C      SUM AIRCRAFT OVER ALL SECTOR CONTAINED IN SAME REGION AS SECTOR
C      IS FOR ATTACKER
C
C      JR=NR(L)
C      DO 64005 IR=1, JR
C      IRT=IR+(L-1)*NR(1)
C      JLS=NLSR(IRT)
C      JHS=NHSR(IRT)
C
C      DO 64010 KS=JLS, JHS
C      SUM=SUM+FAAAFZ(IAC, IR, L)*ACFS(IAC, KS, L)
C
C      SUM1=SUM1+FAAARZ(IAC, IR, L)*ACRS(IAC, KS, L)
64010  CONTINUE
64005  CONTINUE
C      SUM2=FAAAZZ(IAC, L)*ACCZ(IAC, L)                                M JC1001
```

Error D17: Incorrect parameter used for reconnaissance aircraft attrition.

Location: In subroutine TC within DO loop 20.

Discussion: The local parameter SUMM(IM) contains the probability that a given type of army reconnaissance aircraft is attrited while on a mission of type IM in a given cycle. However, one of the parameters used to determine SUMM(IM) is RACAM(IS,IM,L) instead of the correct RAACM(IAAC,IM,L) that gives the rate of attrition for army reconnaissance aircraft type IAAC on mission of type IM for side L.

Corrective Action: In the assignment statement of SUMM(IM), substitute the parameter RAACM(IAAC,IM,L) for the parameter RACAM(IS,IM,L).

Original:

```
C      COMPUTE ATTRITION AND REMAINING INVENTORIES TO ARMY AIR CARRIERS.
C      DO 30 L=1,2
C      N3=NAAC(L)
C      DO 25 IS=1,NS
C      DO 25 IAAC=1,N3
C      DO 20 IM=1,3
C      SUMM(IM) = PAACAM(IAAC,IM,L)*RACAM(IS,IM,L)
C                           =(1.0-FRAAC(Iaac,L))
C      X
C      20 CONTINUE
```

Error D17 (Cont'd)

Corrected:

```
C COMPUTE ATTRITION AND REMAINING INVENTORIES TO ARMY AIR CARRIERS.  
DO 30 L=1,2  
N3=NAAC(L)  
DO 25 IS=1,NS  
DO 25 IAAC=1,N3  
DO 20 IM=1,3  
SUMM(IM)=PAACAM(IAAC,IM,L)*RAACM(IAAC,IM,L)*(1.0-FRAAC(Iaac,L)) C JC1001  
20 CONTINUE
```

Error D18: Incorrect determination of COMMZ reconnaissance aircraft attrition.

Location: In subroutine TC following DO loop 42.

Discussion: The assignment statement for parameter RACCZ(L) that contains the updated number of COMMZ reconnaissance aircraft (after current attrition) is incorrectly placed inside DO loop 48, which is indexed over the regions IR for side L. Instead, a new local parameter, say TEMP4, should be utilized to accumulate the average probability that a reconnaissance aircraft is attrited:

$$\text{TEMP4} = \sum_{\text{IR}} \text{TEMP2} * \text{NI/NS},$$

where TEMP2 (already determined) is the average probability that a reconnaissance aircraft is attrited in region IR during a cycle, NI is the number of sectors in the region IR, and NS is the total number of sectors.

Corrective Action: Accumulate the new local parameter TEMP4 within DO loop 48 over all regions IR. Then, following DO loop 48, determine the number of COMMZ reconnaissance aircraft surviving; that is,

$$\text{RACCZ}(L) = \text{AMAX1}[0.0, \text{RACCZ}(L)*(1.-\text{TEMP4})]$$

Original:

```
C SUM IS TOTAL NO. RECON. A/C DESTROYED IN REGION  
SUM=C.  
DO 40 IM=1,3  
DO 40 IS=IS1,IS2  
SUM = SUM + RAAFRM(IM,L)*RACAH(IS,IM,L)*TEMP  
40 CONTINUE  
TEMP2=SUM/TOT  
TEMP3=1.-TEMP2  
C REDUCE INVENTORIES IN PROPORTION TC PERCENTAGE OF A/C DESTROYED.  
DO 42 IS=IS1,IS2  
RACFS(IS,L)=AMAX1(0.0,RACFS(IS,L)*TEMP3)  
RACRS(IS,L)=AMAX1(0.0,RACRS(IS,L)*TEMP3)  
42 CONTINUE  
RACCZ(L)=AMAX1(0.0,RACCZ(L)*(1.-TEMP2*FLOAT(NI)/FLOAT(NS)))  
48 CONTINUE  
50 CONTINUE
```

Error D18 (Cont'd)

Corrected:

```
DO 50 L=1,2
TEMP=SRRAC(L)
TEMP4=0.
N1=1+N2
N2=NR(L)+N2
C LOOP ON SIDE L REGIONS
DO 48 IR=N1,N2
IS1=NLSR(IR)
IS2=NHSR(IR)
C TOT IS TOTAL NO. RECONNAISANCE A/C IN REGION PLUS PROPORTION OF
C RECON. A/C IN COMMZ
TOT=0.
DO 35 IS=IS1,IS2
35 TOT=TOT+RACFS(IS,L)+RACRS(IS,L)
NI=IS2-IS1+1
TOT=TOT+(FLOAT(NI)/FLOAT(NS))*RACCZ(L)
IF(TOT.LE..0001)GO TO 48
C SUM IS TOTAL NO. RECON. A/C DESTROYED IN REGION
SUM=0.
DO 40 IM=1,3
DO 40 IS=IS1,IS2
SUM = SUM + RAAFRM(IM,L)*RACAM(IS,IM,L)*TEMP
40 CONTINUE
TEMP2=SUM/TOT
TEMP3=1.-TEMP2
C REDUCE INVENTORIES IN PROPORTION TO PERCENTAGE OF A/C DESTROYED.
DO 42 IS=IS1,IS2
RACFS(IS,L)=AMAX1(0.0,RACFS(IS,L)*TEMP3)
RACRS(IS,L)=AMAX1(0.0,RACRS(IS,L)*TEMP3)
42 CONTINUE
TEMP4=TEMP4+TEMP2*FLUAT(NI)/FLOAT(NS)
48 CONTINUE
RACCZ(L)=AMAX1(0.0,RACCZ(L)*(1.-TEMP4))
50 CONTINUE
```

I JCI001

I JCI001

C JCI001

Error D19: Complex errors involving updating division locations.

Location: In subroutine TC within DO loop 2050.

Discussion: Three major errors exist with DO loop 2050:

(a) Local parameter N3 is defined in the first part of the DO loop as the lower index limit of the division identification index for the defending side. However, this parameter is redefined later in the DO loop for other purposes, then used in DO loop 2061 for its original purpose.

(b) For the defending side, divisions located within the old 1st inactive battle area are transferred to the new 1st inactive battle area (which is identical to the old 2nd inactive battle area). Later, for the defending side, divisions located in the old 2nd inactive battle area are transferred to the new 2nd inactive battle area. However, these latter divisions now erroneously include the former.

Error D19 (Cont'd)

Discussion (Cont'd)

(c) The code does not allow for the case in which the FEBA change is so great that the new active battle area is not an adjacent battle area to the old active battle area.

Corrective Action: (a) Use a new local parameter N5 instead of N3 as the limiting index in DO loops 2010, 2015, 2025, and 2030.

(b) For the defending side, allow divisions in the old 2nd inactive division to retreat before the divisions in the old 1st inactive division.

(c) Allow a loop back to check whether the new FEBA lies beyond the boundary of the new active battle area.

Original:

```
C -----
C 20) UPDATE DIVISION LOCATION DUE TO FEBA MOVEMENT.
C -----
C      DD 2050 IS=1,NS
C      SAVE OLD ACTIVE BATTLE AREA LOCATIONS FOR ONE CYCLE
C      IABASP(IS) = IABAS(IS)
C      IBA=IABAS(IS)
C      IF(FEBA(IS).LT.GDBA(IBA)) GO TO 2005
C      RED SIDE IS ADVANCING
C      ISIGN=1
C      L=1
C      K=2
C      N1=ND(1)+MAD(1)+1
C      N2=ND(1)+MAD(1)+ND(2)
C      N3 = 1
C      N4 = ND(1)
C      GJ TU 2008
2005  IBA1=IBA-NS
C      IF FEBA HAS ADVANCED PASSED FIRST BATTLE AREA IN SECTOR, SKIP
C      IF(IBA1.LE.0) GO TO 2050
C      IF FEBA STILL IN ACTIVE BATTLE AREA, SKIP
C      IF(FEBATIS).GT.GDBA(IBA1)) GU TO 2050
C      BLUE SIDE IS ADVANCING
C      ISIGN=-1
C      L=2
C      K=1
C      N1=1
C      N2=NL(1)
C      N3 = ND(1) + MAD(1) + 1
C      N4 = ND(1) + ND(2) + MAD(1)
2003  IBAF=IBA+ISIGN-NS
C      IF FEBA OUTSIDE THEATER, SKIP
C      IF(IBAF.GT.NBA.OR.IBAF.LE.0) GO TO 2050
C      SET NEW ABA AND SUPPLY NODE SERVING IT.
C      IABAS(IS)=IBAF
C      ISNBA(IBAF)=ISNABA(IS,K)
C      MOVE DIVISION FROM OLD ABA TO NEW ABA
C      N3=NDS(IS,1)+NDS(IS,2)
C      DD 2010 IDS=1,N3
C      ID=IDLABA(IDS,IS)
```

Error D19 (Cont'd)

Original (Cont'd)

```
IBALD(ID)=IBAF
2010 CONTINUE
C MOVE RETREATING DIVISIONS FROM OLD ABA TO NEW 1ST INACTIVE BA-
IBA2=IBAF+ISIGN=NS
N3=NDIBA(IS,L)
IF(N3.EQ.0) GO TO 2060
IF(IBA2.LE.0.OR.IBA2.GT.NBA) GO TO 2020
C SET NEW LOCATIONS
DO 2015 IDV=1,N3
ID=IDLIBA(IDV,IS,L)
IBALD(ID)=IBA2
2015 CONTINUE
GO TO 2060
C THERE IS NO NEW 1ST INACTIVE BA FOR DEFENDER - REMOVE DIVISIONS
C FROM THEATER.
2020 DU 2025 IDV=1,N3
ID=IDLIBA(IDV,IS,L)
IBALD(ID)=0
2025 CONTINUE
NDIBA(IS,L)=0
C MOVE DEFENDERS IN SECOND INACTIVE BATTLE AREA BACK
2060 CONTINUE
IBA3 = IBA2+ISIGN=NS
IF(IBA3.LE.0.OR.IBA3.GT.NBA) GO TO 2063
DO 2061 ID=N3,N4
IF(IBALD(ID).NE.IBA2) GO TO 2061
IBALD(ID) = IBA3
2061 CONTINUE
2063 CONTINUE
2030 N3=NDIBA(TS,K)
IF(N3.EQ.0)GO TO 2037
DO 2035 IDV=1,N3
ID=IDLIBA(IDV,IS,K)
IBALD(ID)=IBA
2035 CONTINUE
C MOVE ADVANCING DIVISIONS FROM OLD 2ND INACTIVE TO NEW 2ND INACTIVE
2037 ISIGN=-ISIGN
IBA2=IBA+ISIGN=NS
IBA3=IBA2+ISIGN=NS
IF(IBA3.LE.0.OR.IBA3.GT.NBA) GO TO 2050
DO 2040 ID=N1,N2
IF(IBALD(ID).NE.IBA3) GO TO 2040
IBALD(ID)=IBA2
2040 CONTINUE
2050 CONTINUE
```

Error D19 (Cont'd)

Corrected:

C-----
C 201 UPDATE DIVISION LOCATION DUE TO FEBA MOVEMENT.
C-----
C DO 2050 IS=1,NS
C SAVE OLD ACTIVE BATTLE AREA LOCATIONS FOR ONE CYCLE
IABASP{IS} = IABAS{IS} C JC1001
2002 IBA=IABASE{IS}
IF{FEBA{IS}.LE.GDBA{IBA}} GO TO 2005
C RED SIDE IS ADVANCING
ISIGN=1
L=1
K=2
N1=ND{1}+MAD{1}+1
N2=ND{1}+MAD{1}+ND{2}
N3 = 1
N4 = ND{1}
GO TO 2008
2005 IBAI=IBA-NS
TEMP=0.
IF{IBAI.LE.0} GO TO 2007
TEMP=GDBA{IBA1} I JC1001
C IF FEBA STILL IN ACTIVE BATTLE AREA, SKIP C JC1001
2007 IF{FEBA{IS}.GT.TEMP} GO TO 2050 C JC1001
C BLUE SIDE IS ADVANCING
ISIGN=-1
L=2
K=1
N1=1
N2=ND{1}
N3 = ND{1} + MAD{1} + 1
N4 = ND{1} + ND{2} + MAD{1}
2008 IBAF=IBA+ISIGN*NS
C IF FEBA OUTSIDE THEATER, SKIP
IF{IBAF.GT.NBA.DR.IBAF.LE.0} GO TO 2050
C SET NEW ABA AND SUPPLY NODE SERVING IT.
IABAS{IS}=IBAF
ISNBA{IBA}=ISNABA{IS,K}
C... MOVE DIVISIONS FROM OLD ABA TO NEW ABA
NS=NDS{IS,1}+NDS{IS,2} C JC1001
DO 2010 IDS=1,N5 C JC1001
ID=IDLABA{IDS,IS}
IBALD{ID}=IBAF
2010 CONTINUE
C.... MOVE RETREATING DIVISIONS FROM OLD 2ND TO NEW 2ND INACTIVE BA.1 JC1001
IBA2=IBAF+ISIGN*NS
IBA3 = IBA2+ISIGN*NS M JC1001
IF{IBA3.LE.0.DR.IBA3.GT.NBA} GO TO 2063 M JC1001
DO 2061 ID=N3,N4 M JC1001
IF{IBALD{ID}.NE.IBA2} GO TO 2061 M JC1001
IBALD{ID} = IBA3 M JC1001
2061 CONTINUE M JC1001
2063 CONTINUE M JC1001

Error D19 (Cont'd)

Corrected (Cont'd)

C.... MOVE RETREATING DIVISIONS FROM OLD 1ST TO NEW 1ST INACTIVE BA.
N5=NDIBA{IS,L} C JCI001
IF(N5.EQ.0) GO TO 2060 C JCI001
IF(IBA2.LE.0.OR.IBA2.GT.NBA) GO TO 2020
C SET NEW LOCATIONS
DO 2015 IDV=1,N5 C JCI001
ID=IDLIBA(IDV,IS,L)
IBALD(ID)=IBA2
2015 CONTINUE
GO TO 2060
C THERE IS NO NEW 1ST INACTIVE BA FOR DEFENDER - REMOVE DIVISIONS
C FROM THEATER.
2020 DO 2025 IDV=1,N5 C JCI001
ID=IDLIBA(IDV,IS,L)
IBALD(ID)=0
2025 CONTINUE
NDIBA{IS,L}=0
2060 CONTINUE
C.... MOVE ADVANCING DIVISIONS FORM OLD 1ST TO NEW 1ST INACTIVE BA. I JCI001
2030 N5=NDIBA{IS,K} C JCI001
IF(N5.EQ.0) GO TO 2037 C JCI001
DO 2035 IDV=1,N5 C JCI001
ID=IDLIBA(IDV,IS,K)
IBALD(ID)=IBA
2035 CONTINUE
C.... MOVE ADVANCING DIVISIONS FROM OLD 2ND TO NEW 2ND INACTIVE BA.
2037 ISIGN=-ISIGN C JCI001
IBA2=IBA+ISIGN*NS
IBA3=IBA2+ISIGN*NS
IF(IBA3.LE.0.OR.IBA3.GT.NBA) GO TO 2045
DO 2040 ID=N1,N2 C JCI001
IF(IBALD(ID).NE.IBA3) GO TO 2040
IBALD(ID)=IBA2
2040 CONTINUE
2045 CONTINUE I JCI001
GOTO 2002 I JCI001
2050 CONTINUE

Error D20: Parameter SACFRB(IS,2,L) is erroneously reset.

Location: In subroutine TC within DO loop 2072.

Discussion: The parameter SACFRP(IS,2,L) that contains the number of aircraft shelters within rear region of sector IS for side L is reset, and is not reassigned to finite value later.

Corrective Action: Eliminate the statement that resets the parameter SACFRB(IS,2,L).

Original:

DO 2072 L=1,2
DO 2072 IS=1,NS
IF(IABAS{IS}.EQ.IABASP{IS}) GO TO 2072
ABASEF{IS,L} = 0.
SACFRB{IS,1,L} = 0.
SACFRB{IS,2,L} = 0.

2072 CONTINUE

Error D20 (Cont'd)

Corrected:

```
DO 2072 L=1,2
DO 2072 IS=1,NS
IF(IABASIS).EQ.IABASP(IS) GO TO 2072
ABASEF(IS,L) = 0.
SACFRB(IS,1,L) = 0.
C JCI SACFRB(IS,2,L) = 0.
2072 CONTINUE
```

C JC1001

Error D21: DO statement 5503 is missing.

Location: In subroutine TC within DO loop 5550.

Discussion: Within the context of the code, a DO statement, whose index L is ranged over the combatant sides, is missing. The DO label is 5503.

Corrective Action: Insert the DO statement within the DO loop 5550; that is,

DO 5503 L = 1,2

Original:

```
C -----
C 55) WITHDRAW ALL INEFFECTIVE DIVISIONS AND REINFORCE WITH DIVISIONS
C      OF HIGHEST EFFECTIVENESS.
C -----
DO 5550 IS=1,NS
N10=NAAC(L)
DO 5500 IAAC=1,N10
STUR2(IAAC,IS,L)=0.
STUR1(IAAC,IS,L)=ACDS(IAAC,IS,L)
5500 CONTINUE
```

Corrected:

```
C -----
C 55) WITHDRAW ALL INEFFECTIVE DIVISIONS AND REINFORCE WITH DIVISIONS
C      OF HIGHEST EFFECTIVENESS.
C -----
DO 5550 IS=1,NS
N2=0
DO 5503 L=1,2
N10=NAAC(L)
DO 5500 IAAC=1,N10
STUR2(IAAC,IS,L)=0.
STUR1(IAAC,IS,L)=ACDS(IAAC,IS,L)
5500 CONTINUE
```

I JC1001
I JC1001

Error D22: Parameter N2 is not initialized.

Location: In subroutine TC before DO loop 5503.

Discussion: The local parameter N2 is used within DO loop 5503 to form the offset for the division identification index IDS. However, N2 should be initialized to zero before entry into DO loop 5503.

Corrective Action: Initialize N2 to zero before DO loop 5503.

Original:

```
C -----
C 55) WITHDRAW ALL INEFFECTIVE DIVISIONS AND REINFORCE WITH DIVISIONS
C      OF HIGHEST EFFECTIVENESS.
C -----
      DD 555: IS=1,NS
      N10=NAAC(L)
      DD 5500 IAAC=1,N10
      STOR2(IAAC,IS,L)=0.
      STOR1(IAAC,IS,L)=AACDS(IAAC,IS,L)
5500 CONTINUE
```

Corrected:

```
C -----
C 55) WITHDRAW ALL INEFFECTIVE DIVISIONS AND REINFORCE WITH DIVISIONS
C      OF HIGHEST EFFECTIVENESS.
C -----
      DD 5550 IS=1,NS
      N2=0                                I JCI001
      DD 5503 L=1,2                         I JCI001
      N10=NAAC(L)
      DD 5500 IAAC=1,N10
      STOR2(IAAC,IS,L)=0.
      STOR1(IAAC,IS,L)=AACDS(IAAC,IS,L)
5500 CONTINUE
```

Error D23: Incorrect side index.

Location: In subroutine TC following DO loop 5537.

Discussion: The conditional branch statement following DO loop 5537 uses a side index L instead of the correct index 1.

Corrective Action: Change the side index from L to 1.

Original:

```
5537 CONTINUE
C
C      FINALLY RECONSTITUTE IDLABA
C      IF TOO MANY DIVISIONS IN ABA , PROGRAM TERMINATES
C      IF(NDS(IS,L)+NDS(IS,2).LE.MDDABA) GO TO 5539
C      WRITE(MUI,5538)
5538 FURMAT(50H   TEL MANY DIVISIONS ASSIGNED TO A SECTOR IN TC
      STOP 1111
5539 CONTINUE
```

Error D23 (Cont'd)

Corrected:

5537 CONTINUE

C

C FINALLY RECONSTITUTE IDLABA
C IF TOO MANY DIVISIONS IN ABA , PROGRAM TERMINATES
IF(NDS{IS,1}+NDS{IS,2}.LE.MUDABA) GO TO 5539

C JC1001

WRITE(MOT,5538)

5538 FP MAT{50H TOO MANY DIVISIONS ASSIGNED TO A SECTION IN TC
' JP 1111

5539 CONTINUE

Error D24: Incorrect division-type index used.

Location: In subroutine TC within DO loop 6526.

Discussion: Within DO loop 6526 local parameter TEMP2 contains the apportioned number of army reconnaissance aircraft for division ID that is being withdrawn. However, the factor FAACTD(IT1) used to evaluate TEMP2 uses index IT1 for the division type instead of the correct index IT.

Corrective Action: Substitute index IT for index IT1 within the factor FAACTD in the assignment statement for TEMP2.

Original:

C MOVE INEFFECTIVE DIVISION FROM ABA TO SECOND INACTIVE B/A.
6525 IBALD{ID}=IBA
C WITHDRAW ARMY AIR CARRIERS IN PROPORTION TO TDE OF DIV. TO TDE
C OF ALL DIVISIONS IN ABA. STOR1 IS ACTUAL NO. AAC IN ABA.
DO 6526 IAAC=1,N10
IF(STOR2{IAAC,IS,L}.LE.0.0) GO TO 6526
TEMP2=(FAACTD{IT1}*TAACND {IAAC,L})/STOR2{IAAC,IS,L})=
X STOR1{IAAC,IS,L}
AACDS{IAAC,IS,L}=AMAX1(0.0,AACDS{IAAC,IS,L}-TEMP2)
6526 CONTINUE

Corrected:

C MOVE INEFFECTIVE DIVISION FROM ABA TO SECOND INACTIVE B/A.
6525 IBALD{ID}=IBA
C WITHDRAW ARMY AIR CARRIERS IN PROPORTION TO TDE OF DIV. TO TDE
C OF ALL DIVISIONS IN ABA. STOR1 IS ACTUAL NO. AAC IN ABA.
DO 6526 IAAC=1,N10
IF(STOR2{IAAC,IS,L}.LE.0.0) GO TO 6526
TEMP2=(FAACTD{IT}*TAACND {IAAC,L})/STOR2{IAAC,IS,L})=
X STOR1{IAAC,IS,L}
AACDS{IAAC,IS,L}=AMAX1(0.0,AACDS{IAAC,IS,L}-TEMP2)
6526 CONTINUE

Error D25: Parameter IDDABA is undetermined for replacement divisions.

Location: In subroutine TC following statement labelled 6528.

Discussion: When a division labelled ID1 in the 1st inactive battle area replaces an ineffective division in the active battle area, the local parameter IDDABA(ID1) should be set to 1 as an indicator that the replacing division has not been reconstituted.

Corrective Action: Following statement labelled 6528 include the assignment statement

IDDABA(TD1) = 1.

Original:

```
C REMOVE DIVISION FROM INACTIVE LIST
IDLIBA(IKT,IS,L)=0
IBALD(ID1)=IABAS(IS)
IKT = 0
C ADD ARMY AIR CARRIERS FOR NEW DIVISION AT TOE LEVEL
DO 6529 IAAC=1,N10
AACDS(IAC,IS,L)=AACDS(IAC,IS,L)+FAACTD(IT1)*TAACND(IAAC,L)
6529 CONTINUE
```

Corrected:

```
C REMOVE DIVISION FROM INACTIVE LIST
IDLIBA(IKT,IS,L)=0
IDDABA(ID1)=1
IBALD(ID1)=IABAS(IS)                                I JCI001
IKT = 0
C ADD ARMY AIR CARRIERS FOR NEW DIVISION AT TOE LEVEL
DO 6529 IAAC=1,N10
AACDS(IAC,IS,L)=AACDS(IAC,IS,L)+FAACTD(IT1)*TAACND(IAAC,L)
6529 CONTINUE
```

Error D26: Double entry of division reconstitution requirements.

Location: In subroutine TC within DO loop 7015.

Discussion: If a division whose index is ID in the active battle area requires a large reconstitution effort, then the parameter IDDABA(ID) is set to 1 and initial reconstitution is bypassed. However, reconstitution requirements are accumulated. Later, a second reconstitution attempt is made within DO loop 7015 and the reconstitution requirements are again accumulated, thus causing a possible double entry in the accumulation process.

Corrective Action: Eliminate the conditional branch statement within DO loop 7005, thus allowing an unconditional reset of the accumulation parameters RWNABA and RPNAFA.

Error D26 (Cont'd)

Original:

```
C.....IF DEMAND FOR REPLACEMENTS HAS BEEN MET, SKIP.  
N3=NW(L)  
DO 7015 IDS=N1,N2  
ID=IDLABA(IDS,IS)  
IF(IUDABA(ID).NE.1.AND.IRDABA(L).NE.1) GO TO 7015  
IT=ITD(ID)  
IC=CNTRYC(ID)  
TEMP=AMAX1(0.0,TPD(IT)-PDIV(ID))  
TEMPA = TPD(IT)*FTDSTR(1,L)  
TEMP = AMIN1(TEMP,TEMPA)  
RPNABA(IC,L)=RPNABA(IC,L)+TEMP  
DU 7010 IW=1,N3  
TEMP=AMAX1(0.0,TWD(IW,IT)-WDIV(IW, ID))  
RWNA(B(IW, IC, L)=RWNA(B(IW, IC, L)+TEMP  
7010 CONTINUE  
7015 CONTINUE
```

Corrected:

```
C 701 COMPUTE DEMAND AND REPLACEMENTS IN THE ACTIVE BATTLE AREA  
C -----  
C DO 7005 L=1,2  
C.....IF DEMAND FOR REPLACEMENTS HAS BEEN MET, SKIP.  
C JCI IF(IRDABA(L).EQ.0) GO TO 7005 C JC1001  
N3=NW(L)  
DO 7004 IC=1,1  
DO 7003 IW=1,N3  
C RWNA(B IS REPLACEMENT WEAPONS NEEDED IN ABA  
7003 RWNA(B(IW, IC, L)=0.  
C RPNABA IS REPLACEMENT PEOPLE NEEDED IN ABA  
RPNABA(IC, L)=0.  
7004 CONTINUE  
7005 CONTINUE  
C FOR EACH DIVISION IN AN ABA DETERMINE REPLACEMENT WEAPONS  
C AND PEOPLE TO BRING DIVISIONS UP TO TOE LEVELS.  
DO 7025 IS=1,NS  
N2=0  
DO 7020 L=1,2  
IF(NDS(IS,L).EQ.0) GO TO 7020  
N1=1+N2  
N2=N2+NDS(IS,L)  
C.....IF DEMAND FOR REPLACEMENTS HAS BEEN MET, SKIP.  
N3=NW(L)  
DO 7015 IDS=N1,N2  
ID=IDLABA(IDS,IS)  
IF(IUDABA(ID).NE.1.AND.IRDABA(L).NE.1) GO TO 7015  
IT=ITD(ID)  
IC=CNTRYC(ID)  
TEMP=AMAX1(0.0,TPD(IT)-PDIV(ID))  
TEMPA = TPD(IT)*FTDSTR(1,L)  
TEMP = AMIN1(TEMP,TEMPA)  
RPNABA(IC,L)=RPNABA(IC,L)+TEMP  
DO 7010 IW=1,N3  
TEMP=AMAX1(0.0,TWD(IW,IT)-WDIV(IW, ID))  
RWNA(B(IW, IC, L)=RWNA(B(IW, IC, L)+TEMP  
7010 CONTINUE  
7015 CONTINUE
```

Error D27: Division-type index is undefined.

Location: In subroutine TC within DO loop 8560.

Discussion: Within DO loop 8560 the assignment statements for local parameters SUNDIV, TPNDIV, and TWNDIV contain parameters having division type index IT. However, index IT is defined in the section following the conditional branch statement 8555.

Corrective Action: Move the assignment statement for index IT to the section following the DO statement 8585 and before the conditional branch to statement 8555.

Original:

```
C.....IF REPLACEMENT POOLS SUFFICIENT, SEND UP REPLACEMENTS.  
NS=NSU(L)  
N6=N1WSU(L)  
DO 8585 ID=N1,N2  
IBA = IBALD(ID)  
IS = MCD(IBA,NS)  
IF (IS.EQ.0) IS=NS  
IABA = TABAS(IS)  
IBAU = IABA+ISIGN*ITYP*NS  
IF (IBA.NE.IBAU) GO TO 8555  
IF (IBDABA(ID).EQ.0.AND.IT.EQ.0) GO TO 8555  
IT=ITD(ID)  
IC=CNTTRYC(ID)  
PDIV(IID)=PDIV(IID)+PNDABA(ID)  
RPCZ(IC,L)=AMAX1(0.0,RPCZ(IC,L)-PNDABA(ID))  
DO 8540 IW=1,N3  
WDIV(IW, ID)=WDIV(IW, ID)+WNDBALIW, ID)  
RPCZ(IW, IC, L)=AMAX1(0.0,WRPCZ(IW, IC, L)-WNDBALIW, ID))  
8541 CONTINUE  
DO 8545 ISU=1,NS  
NSUDT(ISU, ID)=NTSUDT(ISU, ID)  
8545 CONTINUE  
( FILE CAPACITY LIMITED. RESUPPLY AS IN SECTION 75  
GO TO 8585  
8555 CONTINUE
```

Corrected:

```
C.....IF REPLACEMENT POOLS SUFFICIENT, SEND UP REPLACEMENTS.  
NS=NSU(L)  
N6=N1WSU(L)  
DO 8585 ID=N1,N2  
IT=ITD(ID) M JCI001  
IBA = IBALD(ID)  
IS = MCD(IBA,NS)  
IF (IS.EQ.0) IS=NS  
IABA = TABAS(IS)  
IBAU = IABA+ISIGN*ITYP*NS  
IF (IPAU.GT.NBA.DR. IRAU.LE.0) GOTO 8585 T JCI001  
IF (IBA.NE.IBAU) GO TO 8585  
IF (IBDABA(ID).NE.0.DR.IT.NE.0) GU TO 8555 C JCI001  
IC=CNTTRYC(ID)  
PDIV(IID)=PDIV(IID)+PNDABA(ID)
```

Error D27 (Cont'd)

Corrected (Cont'd)

```
RPCZ(IC,L)=AMAX1(0.0,RPCZ(IC,L)-PNDABA(ID))
DD 540 IW=1,N3
WDIV(IW,ID)=WDIV(IW,ID)+WNDABA(IW,ID)
WRPCZ(IW,IC,L)=AMAX1(0.0,WRPCZ(IW,IC,L)-WNDABA(IW,ID))
8540 CONTINUE
DD 545 ISU=1,N5
NSU,L(1SU,1D)=NTSJDT(1SU,1T)
3545 C'N' TUE
C  PULL CAPACITY LIMITED. RESUPPLY AS IN SECTION 75
G'N' T. E565
8555 C'N' TUE
```

Error D28: Incorrect conditional terms in branch statement.

Location: In subroutine TC before DO loop 8540.

Discussion: The conditional branch statement to statement labelled 8555 contains the conditional terms

IDDABA(ID).EQ.0.AND.II.EQ.0.

The section beginning at label 8555 is entered if reconstitution resources are insufficient and apportioned resources are required. However, the local flag II is initialized to 0 and is later set to 1 if the resources are limited. Thus, the above conditional term is the negation of the correct term; that is

IDDABA(ID).NE.0.OR.II.NE.0.

Corrective Action: Negate the original conditional term in the branch statement to label 8555.

Original:

```
IABA = IABA(1S)
IBAU = IAFA+ISIGNV(ITYP-MS
IF(IbA.NE.IBAU) GO TO 8565
IF(ILDABA(ID).EQ.0.AND.II.EQ.0) GO TO 8555
IT=ITD(ID)
IC=CNTRYC(ID)
PDIV(ID)=PDIV(ID)+PNDABA(ID)
RPCZ(IC,L)=AMAX1(0.0,RPCZ(IC,L)-PNDABA(ID))
DD 540 IW=1,N3
WDIV(IW,ID)=WDIV(IW,ID)+WNDABA(IW,ID)
WRPCZ(IW,IC,L)=AMAX1(0.0,WRPCZ(IW,IC,L)-WNDABA(IW,ID))
8541 CONTINUE
```

Error D28 (Cont'd)

Corrected:

```
IABA = IABAS{IS}
IBAU = IABA+ISIGN*ITYP*NS
IF{IBAU.GT.NBA.OR.IBAU.LE.0}GOTO 8585
IF{IBA.NE.IBAU} GO TO 8585
IF{IDDABA{ID}.NE.0.OR.II.NE.0} GO TO 8555
IC=CNTRYC{ID}
PDIV{ID}=PDIV{ID}+PNDABA{ID}
RPCZ{IC,L}=AMAX1{0.0,RPCZ{IC,L}-PNDABA{ID}}
DO 8540 IW=1,N3
WDIV{IW, ID}=WDIV{IW, ID}+WNDABA{IW, ID}
WRPCZ{IW, IC, L}=AMAX1{0.0,WRPCZ{IW, IC, L}-WNDABA{IW, ID}}
8540 CONTINUE
```

I JCI001

C JCI001

Error D29: Incorrect index used.

Location: In subroutine TC at statement labelled 8563.

Discussion: The parameter IDWTSU(1,IW,L) should have the indices IDWTSU(1,ISU,L) where ISU is the proper subunit index.

Corrective Action: Change the index IW to ISU in the parameter IDWTSU.

Original:

```
C.....CREATE NEW SUBUNITS. PNSU IS NO. SUBUNITS FOR WHICH THERE ARE
C AVAILABLE REPLACEMENT PEOPLE. WNSU IS NO. FOR WHICH THERE ARE
C PRIMARY WEAPONS AND SUA IS NO. OF SUBUNITS FOR WHICH THERE ARE
C ENOUGH PEOPLE AND PRIMARY WEAPONS. REDUCE REPLACEMENT POOLS TO
C REFLECT CREATION OF NEW SUBUNITS NEEDED IN DIVISION.
SUM=0.
DO 8570 ISU=1,N5
PNSU=0.0
IF(TPN.DIV.LE..001) GO TO 8563
PNSU=(PNDABA{ID}/TPNDIV)*SUNDIV{ISU}
PNSU=AMINI(PNSU,SUNDIV{ISU})
8563 IW=IDWTSU{1,IW,L}
```

Corrected:

```
C.....CREATE NEW SUBUNITS. PNSU IS NO. SUBUNITS FOR WHICH THERE ARE
C AVAILABLE REPLACEMENT PEOPLE. WNSU IS NO. FOR WHICH THERE ARE
C PRIMARY WEAPONS AND SUA IS NO. OF SUBUNITS FOR WHICH THERE ARE
C ENOUGH PEOPLE AND PRIMARY WEAPONS. REDUCE REPLACEMENT POOLS TO
C REFLECT CREATION OF NEW SUBUNITS NEEDED IN DIVISION.
SUM=0.
DO 8570 ISU=1,N5
PNSU=0.0
IF(TPNDIV.LF..001) GO TO 8563
PNSU=(PNDABA{ID}/TPNDIV)*SUNDIV{ISU}
PNSU=AMINI(PNSU,SUNDIV{ISU})
8563 IW=IDWTSU{1,ISU,L}
```

C JCI001

Error D30: DO statement missing.

Location: In subroutine AIRASG before DO Loop 47.

Discussion: Within the context of the code, a DO statement with a DO range 46 indexed over aircraft range IRNG is missing.

Corrective Action: Before DO loop 47 insert the following DO statement

DO 46 IRNG = 1, 5 .

Original:

```
C SOME AIRCRAFT MUST BE REMOVED FROM AN AIRBASE. TRY MOVING SHORTEST
C RANGE AIRCRAFT IN POOL ACROSS SECTORS TO OTHER SECTOR AIRBASES
C THAT MAY HAVE SPACE.
C DO 46 IRNG=1,5
C DO 47 IS=IS1,IS2
C IF(IFLGS(IS).NE.1) GO TO 47
C
C THIS AIRBASE HAS TROUBLE
C DO 48 IAC=1,NACT
C DO ANY AIRCRAFT OF THIS TYPE HAVE TO BE MOVED.
C IF(PPOOL(IAC,IS).EQ.0.) GO TO 48
```

Corrected:

```
C SOME AIRCRAFT MUST BE REMOVED FROM AN AIRBASE. TRY MOVING SHORTEST
C RANGE AIRCRAFT IN POOL ACROSS SECTORS TO OTHER SECTOR AIRBASES
C THAT MAY HAVE SPACE.
C DO 46 IRNG=1,5
C DO 47 IS=IS1,IS2
C IF(IFLGS(IS).NE.1) GO TO 47
C
C THIS AIRBASE HAS TROUBLE
C DO 48 IAC=1,NACT
C DO ANY AIRCRAFT OF THIS TYPE HAVE TO BE MOVED.
C IF(PPOOL(IAC,IS).EQ.0.) GO TO 48
```

C JC1001

Error D31: Incorrect sector index.

Location: In subroutine AIRASG within DO loop 49.

Discussion: The sector index IS is used instead of the correct index ISIN in two parameters within DO loop 49:

- (a) in parameter IFLGS within the conditional branch to label 49;
- (b) in parameter SUMUDR within assignment statement for ACGO.

Corrective Action: Substitute index ISIN for index IS in parameters IFLGS and SUMUDR.

Error D31 (Cont'd)

Original:

```
C NOW SEARCH FOR POTENTIAL RECEPTOR SECTORS
DO 49 ISIN=IS1,IS2
IF (ISIN.EQ.IS) GO TO 49
IF (IFLG5(IS).NE.0) GO TO 49
C NOW ISIN IS A POTENTIAL RECEPTOR AIRBASE. TRANSFER A/C AS POSSIBLE
IF (POOL(IAC,IS).LE.SUMUDR(ISIN)) GO TO 51
C FILL UP RECEPTOR CENTER TO ITS LIMIT
ACGO = SUMUDR(IS)
POOL(IAC,IS) = POOL(IAC,IS)-SUMUDR(ISIN)
SUMUDR(ISIN) = 0.
IFLG5(ISIN) = 2
GO TO 52
51 CONTINUE
```

Corrected:

```
C NOW SEARCH FOR POTENTIAL RECEPTOR SECTORS
DO 49 ISIN=IS1,IS2
IF (ISIN.EQ.IS) GO TO 49
IF (IFLG5(ISIN).NE.0) GO TO 49
C NOW ISIN IS A POTENTIAL RECEPTOR AIRBASE. TRANSFER A/C AS POSSIBLE C JCI001
IF (POOL(IAC,IS).LE.SUMUDR(ISIN)) GO TO 51
C FILL UP RECEPTOR CENTER TO ITS LIMIT C JCI001
ACGO = SUMUDR(ISIN)
POOL(IAC,IS) = POOL(IAC,IS)-SUMUDR(ISIN)
SUMUDR(ISIN) = 0.
IFLG5(ISIN) = 2
GO TO 52
51 CONTINUE
```

Error D32: Incorrect parameter used.

Location: In subroutine AIRASG following statement labelled 170.

Discussion: The context of the code indicates that the parameter SUMAC in the assignment statement

SUMAC(IS) = 0

should be the parameter QPNAC(IS).

Corrective Action: In the assignment statement change the parameter SUMAC to the correct parameter QPNAC.

Error D32 (Cont'd)

Original:

```
173 CONTINUE  
  QPNAC{IS} = 0.9999*PPNOTF{IS,L}/SUMAC{IS}  
174 CONTINUE  
  GO TO 177  
170 CONTINUE  
  SUMAC{IS} = 0.  
177 CONTINUE
```

Corrected:

```
173 CONTINUE  
  QPNAC{IS} = 0.9999*PPNOTF{IS,L}/SUMAC{IS}  
174 CONTINUE  
  GO TO 177  
170 CONTINUE  
  QPNAC{IS} = 0.  
177 CONTINUE
```

C JCI001

DISTRIBUTION

ADMINISTRATOR
DEFENSE TECHNICAL INFORMATION CENTER
DTIC-DDA (12 COPIES)
CAMERON STATION, BUILDING 5
ALEXANDRIA, VA 22314

COMMANDER
US ARMY RSCH & STD GP (EUR)
ATTN CHIEF, PHYSICS & MATH BRANCH
FPO NEW YORK 09510

COMMANDER
US ARMY MATERIEL DEVELOPMENT &
READINESS - RAND
ATTN DRCDE-D, SYSTEMS DEV
ATTN DRCBSI, DIR FOR BATTLEFIELD SYS
INTEGRATION
5001 EISENHOWER AVENUE
ALEXANDRIA, VA 22333

COMMANDER
US ARMY ARMAMENT MATERIEL
READINESS COMMAND
ATTN DRSTAR-LEP-L, TECHNICAL LIBRARY
ATTN DRSTAR-PI, ROBERT E. SCHERTZ
ROCK ISLAND, IL 61299

COMMANDER
US ARMY MISSILE & MUNITIONS
CENTER & SCHOOL
ATTN ATSK-CTD-F
REDSTONE ARSENAL, AL 35809

DIRECTOR
US ARMY MATERIEL SYSTEMS ANALYSIS ACTIVITY
ATTN DRXSY-MP
ABERDEEN PROVING GROUND, MD 21005

DIRECTOR
US ARMY BALLISTIC RESEARCH LABORATORY
ATTN DRDAR-TSB-S (STINFO)
ATTN DRDAR-BLV/MR. D. L. RIGOTTI
ABERDEEN PROVING GROUND, MD 21005

US ARMY ELECTRONICS TECHNOLOGY
& DEVICES LABORATORY
ATTN DELET-DD
FORT MONMOUTH, NJ 07703

HQ USAF/SAMI
WASHINGTON, DC 20330

TELEDYNE BROWN ENGINEERING
ATTN DR. MELVIN L. PRICE, MS-44
CUMMINGS RESEARCH PARK
HUNTSVILLE, AL 35807

ENGINEERING SOCIETIES LIBRARY
ATTN ACQUISITIONS DEPARTMENT
345 EAST 47TH STREET
NEW YORK, NY 10017

DIRECTOR
DEFENSE ADVANCED RESEARCH
PROJECTS AGENCY
ATTN DIR, STRATEGIC TECHNOLOGY OFFICE
ATTN DIR, TACTICAL TECHNOLOGY OFFICE
ARCHITECT BLDG
1400 WILSON BLVD
ARLINGTON, VA 22209

DIRECTOR
DEFENSE COMMUNICATIONS AGENCY
WASHINGTON, DC 20305

DIRECTOR
DEFENSE COMMUNICATIONS AGENCY
COMMAND & CONTROL
TECHNICAL CENTER
ATTN STRAT FORCES DIV
ATTN STRATEGIC OPERATIONS DIV
ATTN DENNIS KONKEL (C315)
WASHINGTON, DC 20301

DIRECTOR
DEFENSE COMMUNICATIONS ENGINEERING CENTER
1860 WIEHLE AVE
RESTON, VA 22090

DIRECTOR
DEFENSE INTELLIGENCE AGENCY
ATTN DT-2, WEAPONS & SYSTEMS DIV
WASHINGTON, DC 20301

CHAIRMAN
OFFICE OF THE JOINT CHIEFS OF STAFF
ATTN J-6, COMMUNICATIONS-ELECTRONICS
ATTN STUDIES ANALYSIS & GAMING AGENCY
WASHINGTON, DC 20301

DIRECTOR
DEFENSE NUCLEAR AGENCY
ATTN F-4EV, ELECTRONICS VULNERABILITY DIV
ATTN STNA, J. T. McDANIEL
WASHINGTON, DC 20305

COMMANDER
FIELD COMMAND
DEFENSE NUCLEAR AGENCY
ATTN FC PRT/MAJ DONALD W. HARD
KIRTLAND AFB, NM 87115

DISTRIBUTION (CONT'D)

UNDER SECRETARY OF DEFENSE
FOR RESEARCH & ENGINEERING
ATTN DEP DIR (TACTICAL
WARFARE PROGRAMS)
ATTN DEP DIR (TEST & EVALUATION)
WASHINGTON, DC 20301

OFFICE OF THE ASSISTANT
SECRETARY OF DEFENSE
PROGRAM ANALYSIS & EVALUATION
ROOM 2E313, THE PENTAGON
WASHINGTON, DC 20301

OFFICE OF THE ASSISTANT
SECRETARY OF DEFENSE
RESEARCH & ENGINEERING (C³I)
ATTN COL DONALD GRIGGS
ROOM 3E163, THE PENTAGON
WASHINGTON, DC 20301

ASSISTANT SECRETARY OF THE ARMY
RES, DEV, & ACQ
ATTN DEP FOR COMMUNICATIONS
& TARGET ACQ
WASHINGTON, DC 20310

OFFICE, DEPUTY CHIEF OF STAFF
FOR OPERATIONS & PLANS
DEPT OF THE ARMY
ATTN DAMO-RQZ, REQUIREMENTS DIR
ATTN DAMO-RQD, COMBAT DIV
ATTN DAMO-TCZ, TELECOM & CMD
& CONTROL DIR
ATTN DAMO-NCN, NUCLEAR DIV
WASHINGTON, DC 20310

ASSISTANT CHIEF OF STAFF
FOR INTELLIGENCE
DEPT OF THE ARMY
WASHINGTON, DC 20310

OFFICE OF THE DEPUTY CHIEF OF STAFF
FOR RESEARCH, DEVELOPMENT,
& ACQUISITION
DEPT OF THE ARMY
ATTN DAMA-RAX, SYSTEMS REVIEW
& ANALYSIS OFFICE
ATTN DAMD-CSZ-A, COMMAND, CONTROL
SURVEILLANCE SYS DIV
ATTN DAMA-CSM, MUNITIONS DIV
ATTN DAMA-WSM, MISSILES & AIR
DEFENSE SYS DIV
ATTN DAMA-CSS-N, NUCLEAR TEAM
ATTN DAMA-WSW, GROUND COMBAT SYS DIV
WASHINGTON, DC 20310

OFFICE, SECRETARY OF THE ARMY
DEPUTY UNDER SECRETARY (OPERATIONS RSCH)
ATTN MR. HARDISON
ROOM 2E614 THE PENTAGON
WASHINGTON, DC 20310

COMMANDER
US ARMY ARMAMENT RESEARCH AND
DEVELOPMENT COMMAND
ATTN DRDAR-LCN, NUCLEAR & FUZE DIV
ATTN DRDAR-LCN-E/WARREN G. REINER
DOVER, NJ 07801

COMMANDER
US ARMY ARMOR CENTER
ATTN TECHNICAL LABORATORY
FORT KNOX, KY 40121

COMMANDER/DIRECTOR
COMBAT SURVEILLANCE
& TARGET ACQUISITION LABORATORY
ATTN DELCS-R/E. L. FROST
FORT MONMOUTH, NJ 07703

COMMANDER
US ARMY COMBINED ARMS COMBA
DEVELOPMENTS ACTIVITY
ATTN ATCA-CIC/MR. L. FOLLIS
FORT LEAVENWORTH, KS 66027

COMMANDER
US ARMY CONCEPTS ANALYSIS AGENCY
ATTN COMPUTER SUPPORT DIV
ATTN METHODOLOGY & RESOURCES DIR
ATTN SYS INTEGRATION ANALYSIS DIR
ATTN WAR GAMING DIR
8120 WOODMONT AVE
BETHESDA, MD 20014

DIRECTOR
ELECTRONIC WARFARE LABORATORY
ATTN DELEW-V/MR. B. MILLER
FT. MONMOUTH, NJ 07703

COMMANDER
US ARMY FOREIGN SCIENCE
& TECHNICAL CENTER
FEDERAL OFFICE BLDG
220 7TH STREET, NE
CHARLOTTESVILLE, VA 22901

COMMANDER
US ARMY INTELLIGENCE & SEC COMMAND
ATTN IAX-OT
ARLINGTON HALL STATION
4000 ARLINGTON BLVD
ARLINGTON, VA 22212

DISTRIBUTION (CONT'D)

COMMANDER
US ARMY MATERIEL SYSTEMS
ANALYSIS ACTIVITY
ATTN DRXSY-C, COMBAT SUPPORT DIV
ATTN DRXSY-DS/MARK RECHES
ATTN DRXSY-GS/B.F. KING
ABERDEEN PROVING GROUND, MD 21005

COMMANDER
US ARMY MOBILITY EQUIPMENT RESEARCH &
DEVELOPMENT COMMAND
ATTN DRDME-RT/DR. KENNETH OSCAR
FORT BELVOIR, VA 22060

DIRECTOR
NIGHT VISION & ELECTRO-OPTICS
LABORATORY
ATTN DELNV-VI/FRANK SHIELDS
FORT BELVOIR, VA 22060

COMMANDER
US ARMY NUCLEAR & CHEMICAL AGENCY
ATTN ATCN-S, STUDIES, ANALYSIS & LIT DIV
ATTN ATCN-W, WEAPONS EFFECTS DIV
7500 BACKLICK RD
BUILDING 2073
SPRINGFIELD, VA 22150

COMMANDER
US ARMY OPERATIONAL TEST
AND EVALUATION AGENCY
ATTN CSTE-ZS, SCI ADVISOR
ATTN CSTE-TM, TEST MANAGERS
5600 COLUMBIA PIKE
FALLS CHURCH, VA 22041

DIRECTOR
US ARMY SIGNALS WARFARE LABORATORY
ATTN DELSW-OS, OFFICE OF THE SCIENTIFIC
ADVISOR
ATTN DELSW-EE, ELECTRONICS/EW DIV
ATTN DELSW-CE, COMM/EW DIV
VINT HILL FARMS STATION
WARRENTON, VA 22186

COMMANDER
US ARMY TRAINING & DOCTRINE COMMAND
ATTN ATCD-N/COL MCJECKI
FORT MONROE, VA 23651

COMMANDER
US ARMY ARMOR SCHOOL
ATTN LIBRARY BR
FORT KNOX, KY 40121

COMMANDANT
US ARMY ENGINEER SCHOOL
ATTN LIBRARY
FORT BELVOIR, VA 22060

COMMANDER
US ARMY FIELD ARTILLERY SCHOOL
ATTN TECH LIBRARY
FORT SILL, OK 73503

COMMANDER
US ARMY INFANTRY SCHOOL
ATTN LIBRARY
FORT BENNING, GA 31905

COMMANDER
US ARMY ORDNANCE CENTER & SCHOOL
ATTN ATSL-CLC-C/PAUL A SCHOLZ
ABERDEEN PROVING GROUND, MD 21005

COMMANDANT
US ARMY SIGNAL CENTER & FT GORDON
ATTN TECH LIBRARY
FORT GORDON, GA 30905

COMMANDER
US ARMY SPECIAL WARFARE SCHOOL
ATTN TECH LIBRARY
FORT BRAGG, NC 28307

DIRECTOR
TRADOC SYSTEMS ANALYSIS ACTIVITY
ATTN ATAA-TDC/MR. C. L. KIRBY
WHITE SANDS MISSILE RANGE, NM 88002

COMMANDER
US ARMY LOGISTIC CENTER
ATTN ATCL-FS/MR. DONALD STEWARDSON
FORT LEE, VA 23801

SUPERINTENDENT
ACADEMY OF HEALTH SCIENCES
ATTN HSA-CDB
FORT SAM HOUSTON, TX 78234

DISTRIBUTION (CONT'D)

US ARMY ELECTRONICS RESEARCH
& DEVELOPMENT COMMAND
ATTN TECHNICAL DIRECTOR, DRDEL-CT
ATTN LEGAL OFFICE

HARRY DIAMOND LABORATORIES
ATTN CO/TD/TSO/DIVISION DIRECTORS
ATTN RECORD COPY, 81200
ATTN HDL LIBRARY, 81100 (3 COPIES)
ATTN HDL LIBRARY, 81100 (WOODBRIDGE)
ATTN TECHNICAL REPORTS BRANCH, 81300
ATTN CHAIRMAN, EDITORIAL COMMITTEE
ATTN CHIEF, 21000
ATTN CHIEF, 21100
ATTN CHIEF, 21200
ATTN CHIEF, 21300
ATTN CHIEF, 21400 (2 COPIES)
ATTN CHIEF, 21500
ATTN CHIEF, 22000
ATTN CHIEF, 22100
ATTN CHIEF, 22300
ATTN CHIEF, 22300
ATTN CHIEF, 22900
ATTN CHIEF, 20240
ATTN MICHALOWICZ, J., 22100
ATTN NOON, T. V., 21200 (10 COPIES)